Numerical Optimization of Computationally Expensive Functions at Argonne National Laboratory

Jeffrey Larson

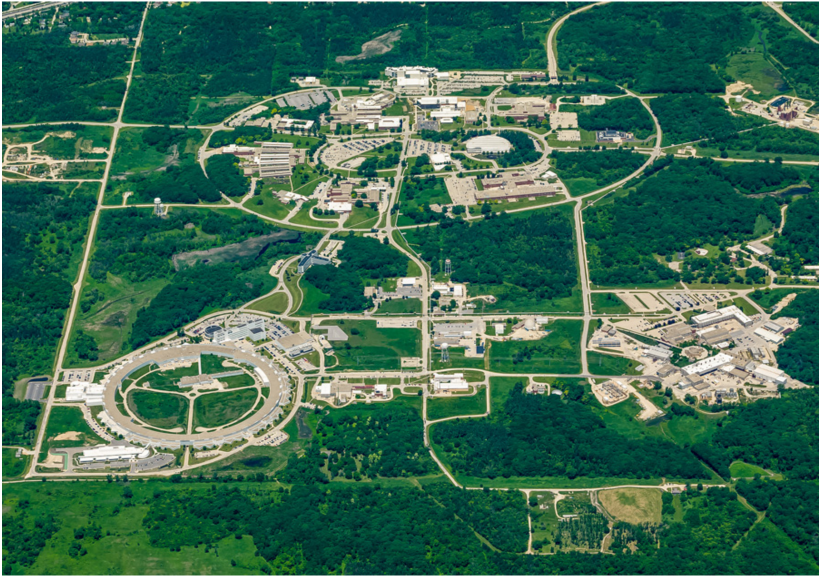
Argonne National Laboratory

October 29, 2018

Argonne National Laboratory

Argonne National Laboratory is a U.S. Department of Energy multidisciplinary science and engineering research center, where researchers work together to answer the biggest questions facing humanity.

Argonne National Laboratory



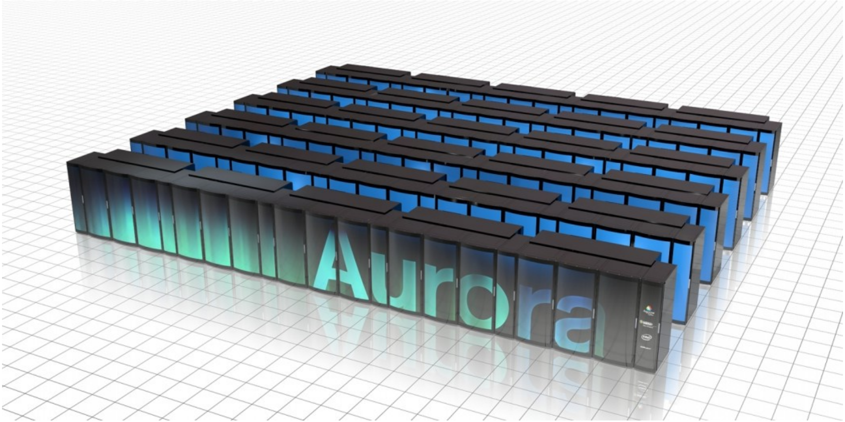
Argonne National Laboratory

Computers/Simulations!

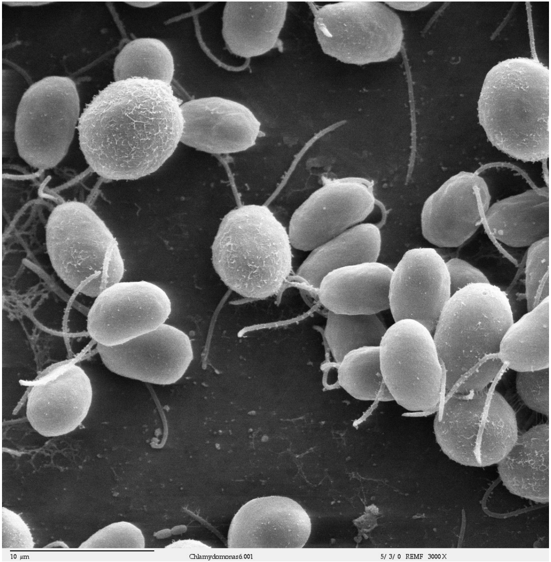
Computers/Simulations!

#17 on TOP500 June 2018 (#6 on TOP500 June 16)

Computers/Simulations!



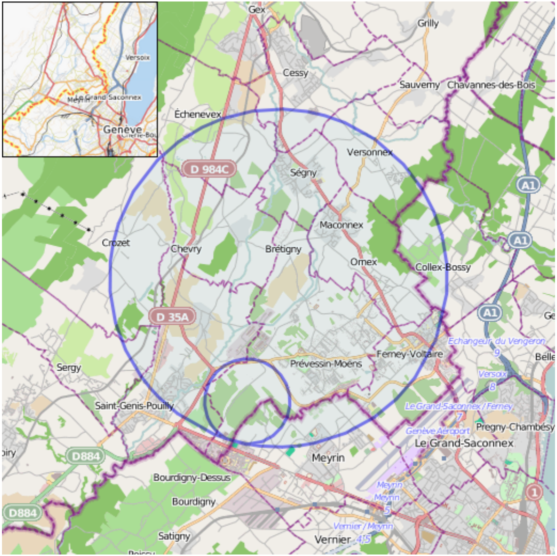
Computers/Simulations!



Dartmouth Electron Microscope Facility

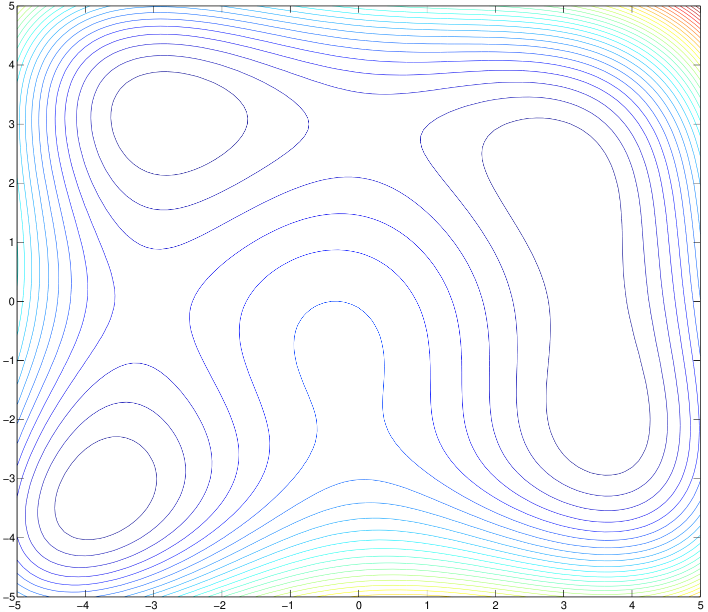
Computers/Simulations!

Computers/Simulations!

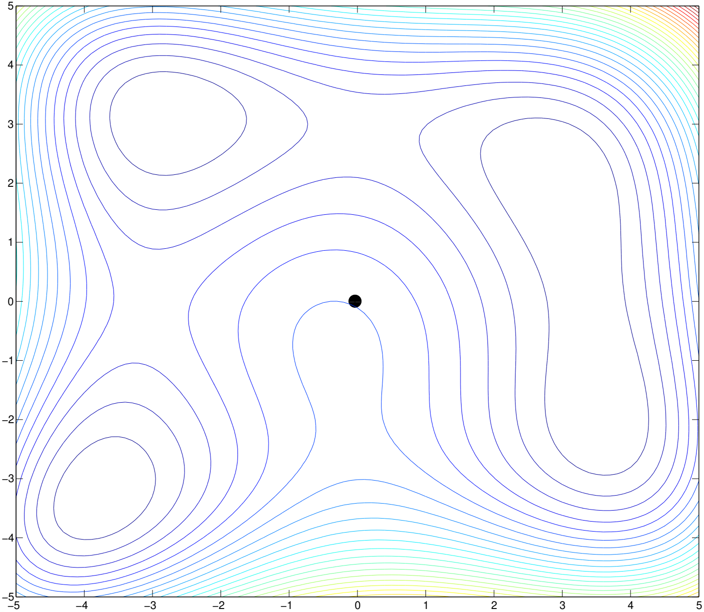


Computers/Simulations!

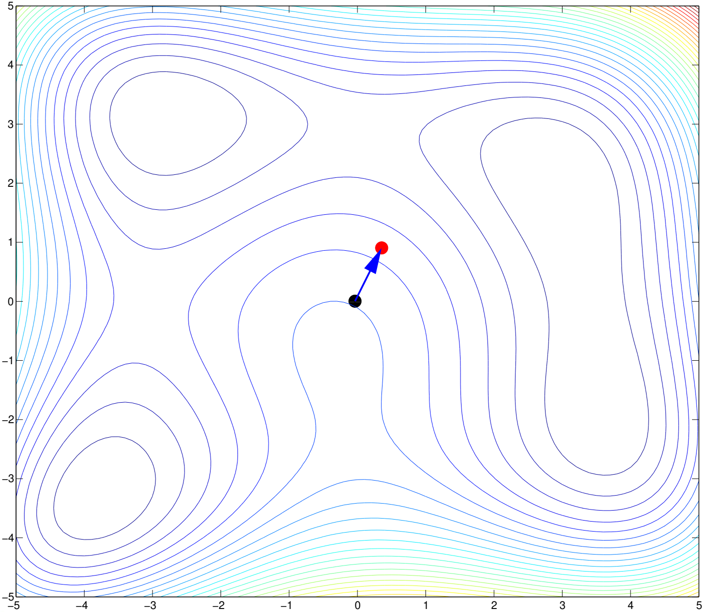
Numerical Optimization - Line Search



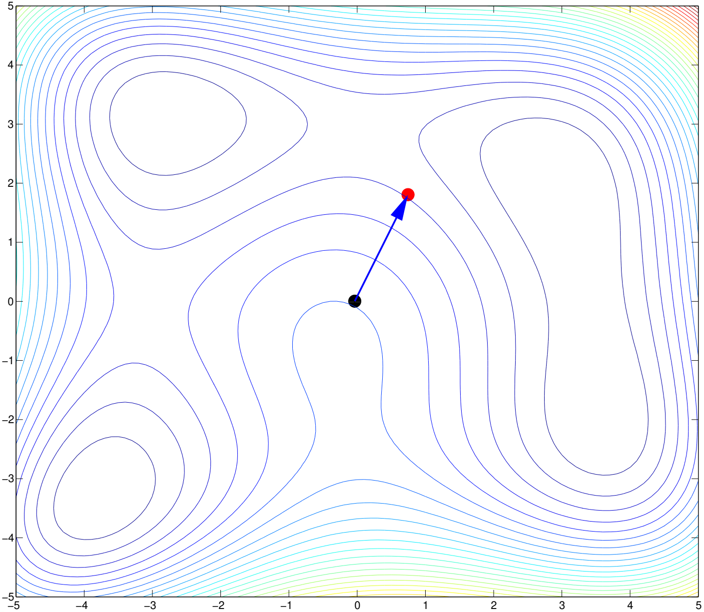
Numerical Optimization - Line Search



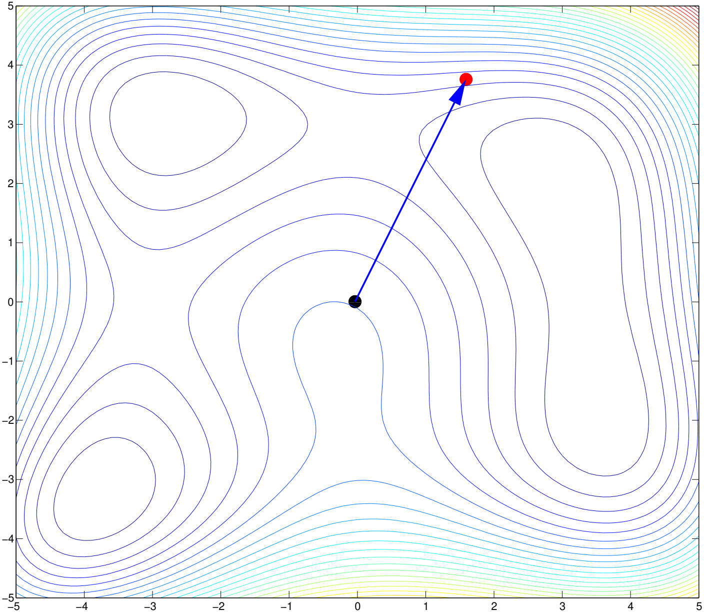
Numerical Optimization - Line Search



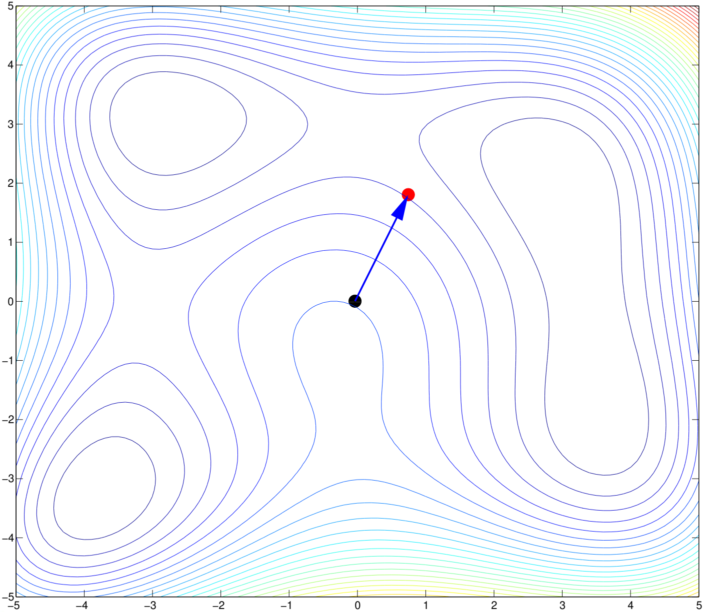
Numerical Optimization - Line Search



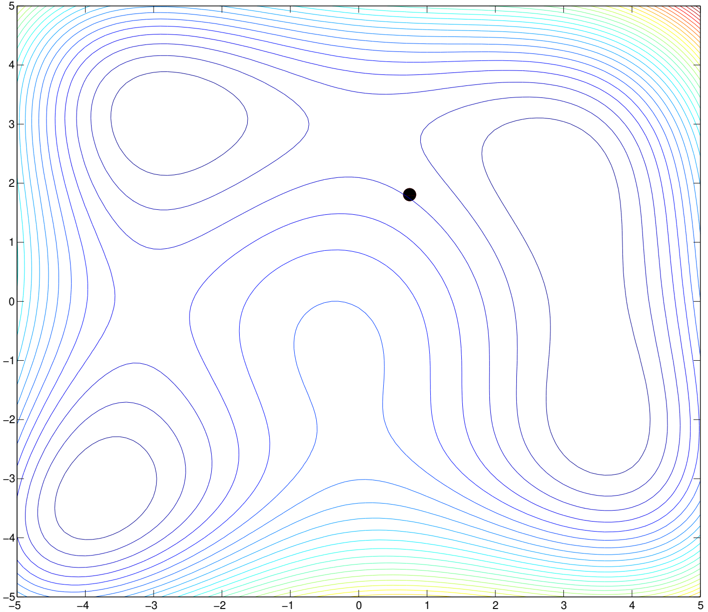
Numerical Optimization - Line Search



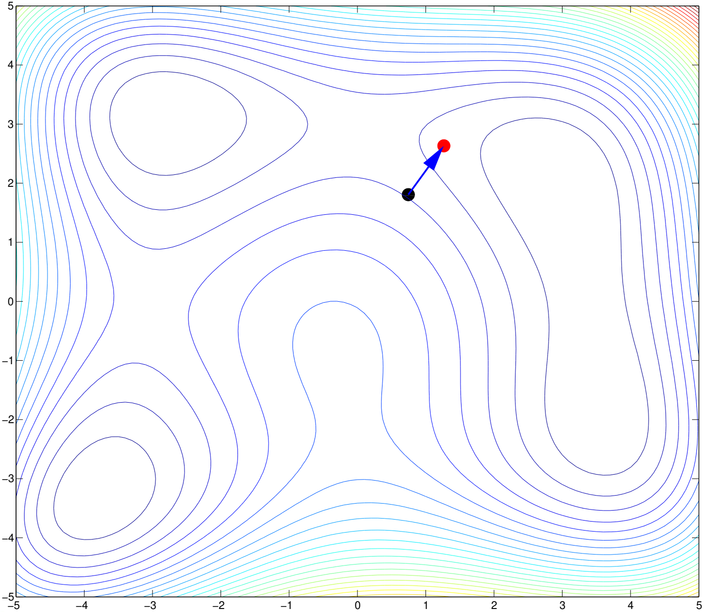
Numerical Optimization - Line Search



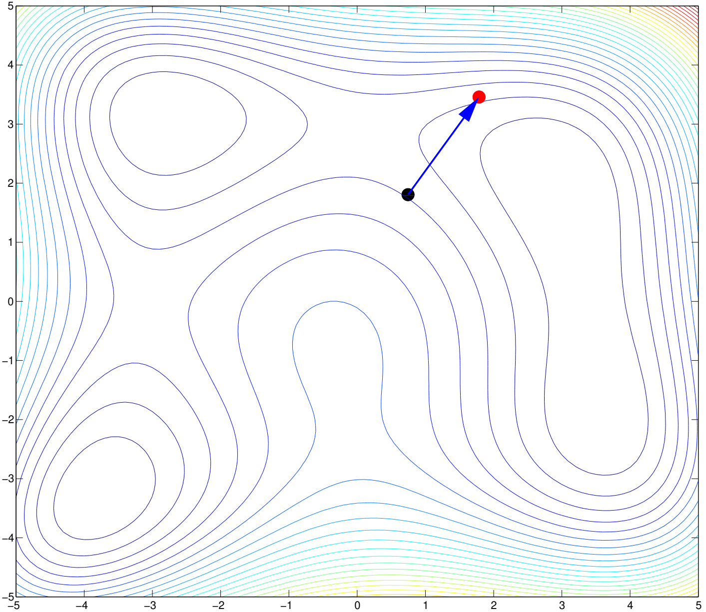
Numerical Optimization - Line Search



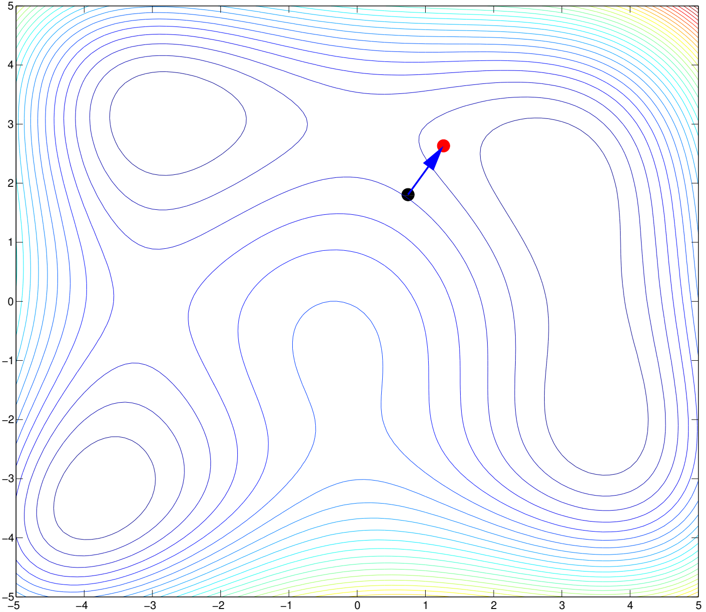
Numerical Optimization - Line Search



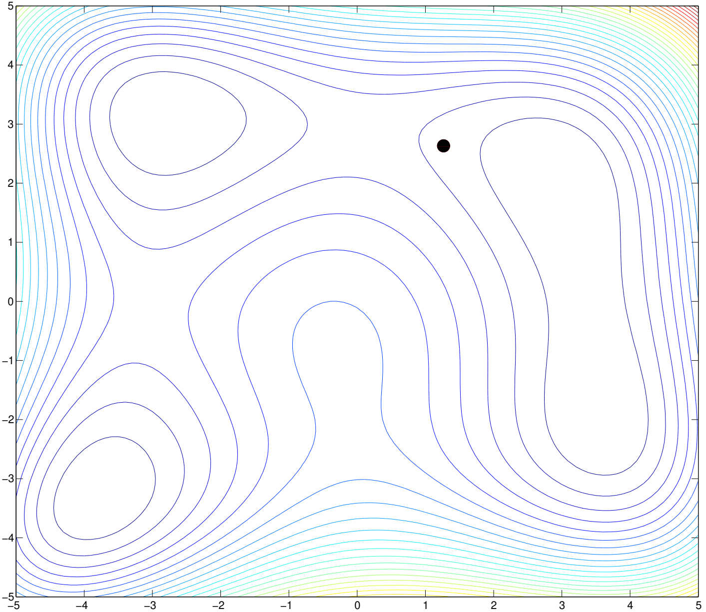
Numerical Optimization - Line Search



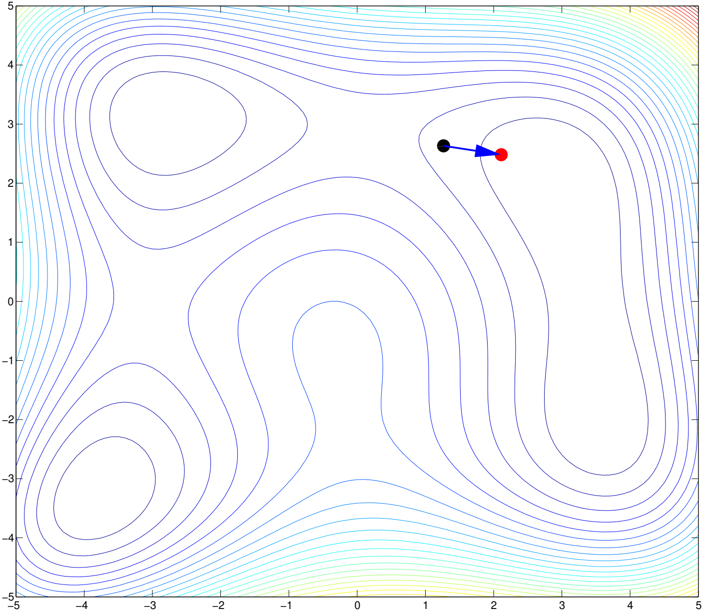
Numerical Optimization - Line Search



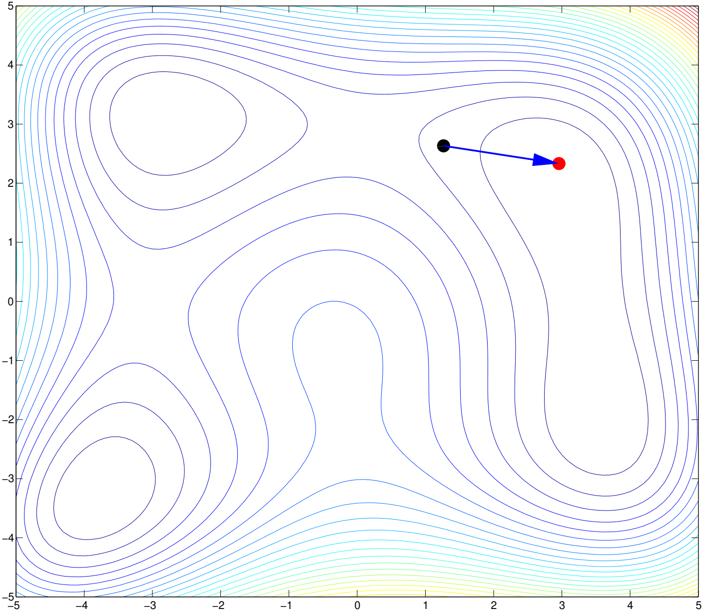
Numerical Optimization - Line Search



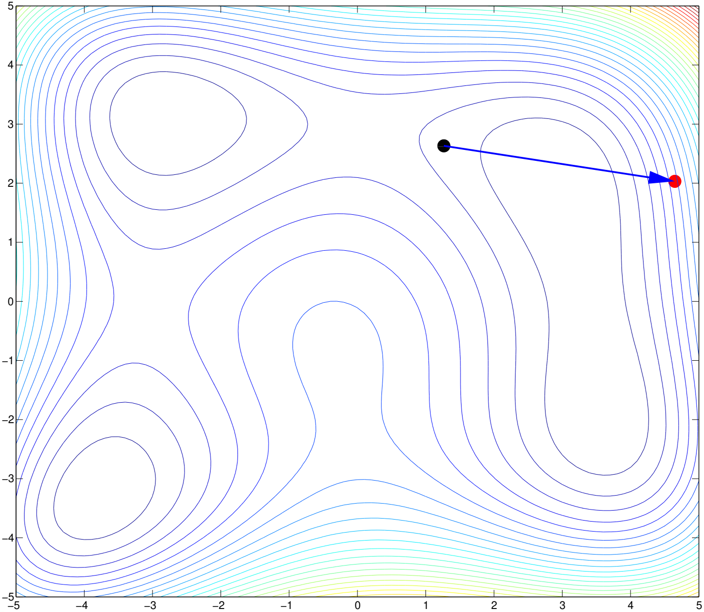
Numerical Optimization - Line Search



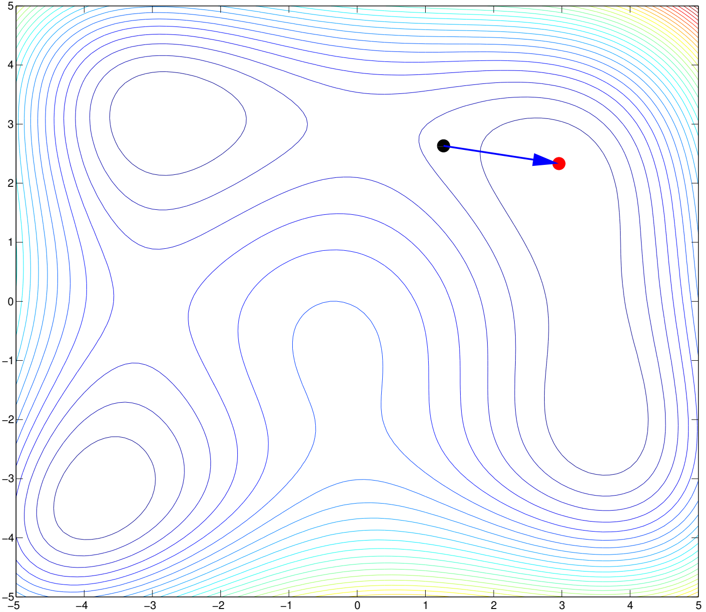
Numerical Optimization - Line Search



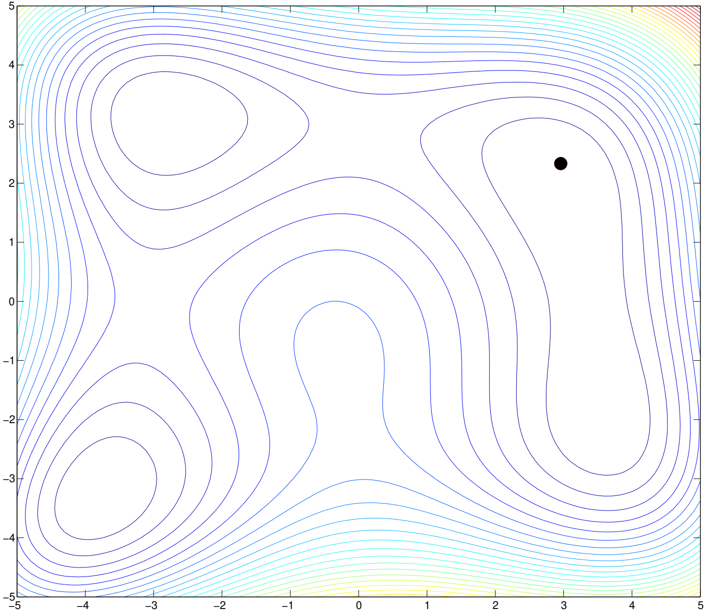
Numerical Optimization - Line Search



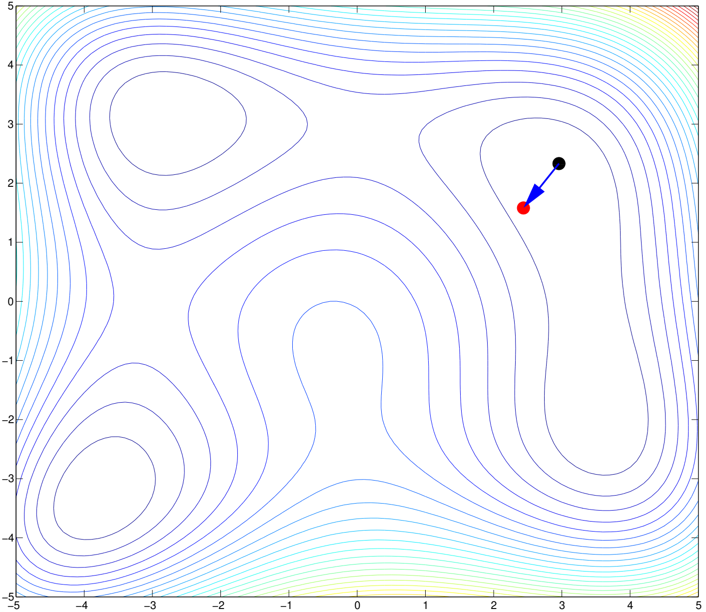
Numerical Optimization - Line Search



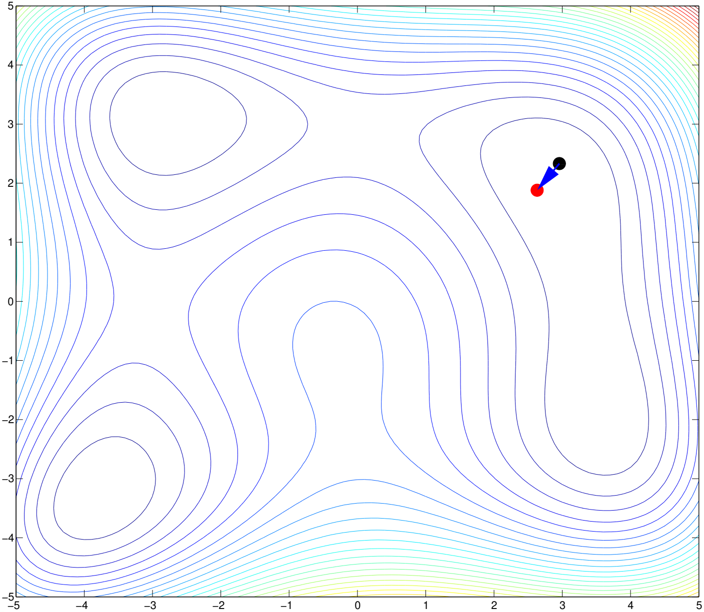
Numerical Optimization - Line Search



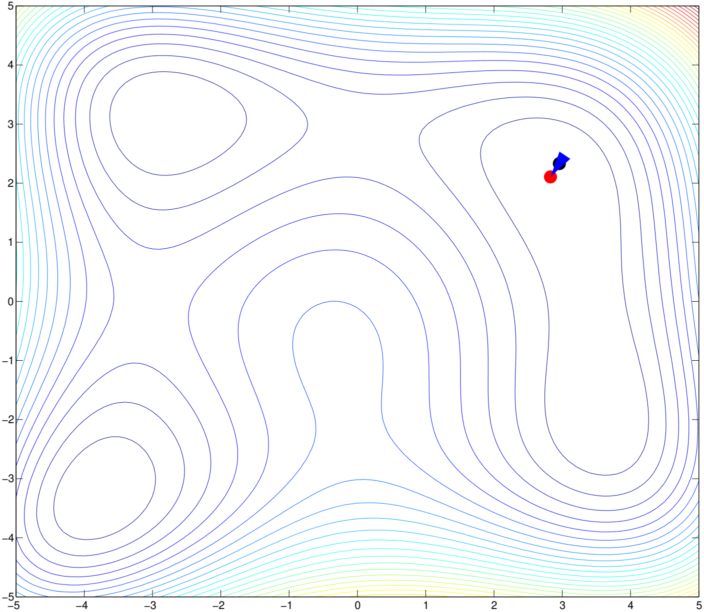
Numerical Optimization - Line Search



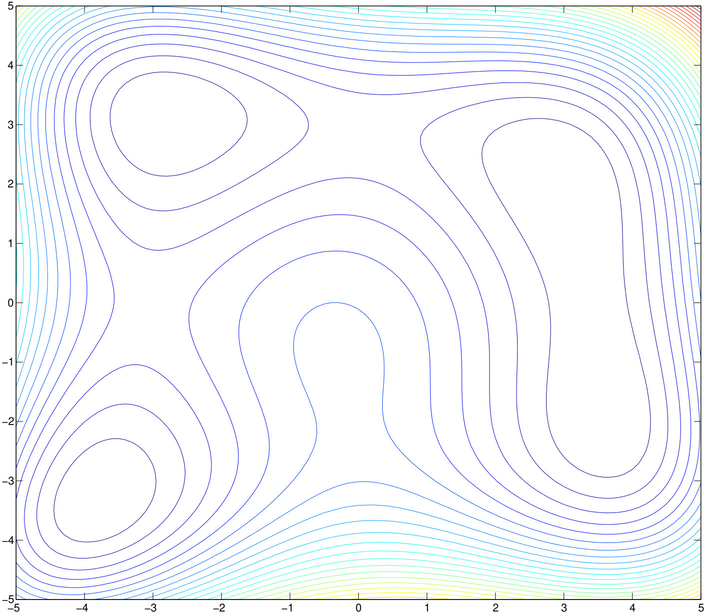
Numerical Optimization - Line Search



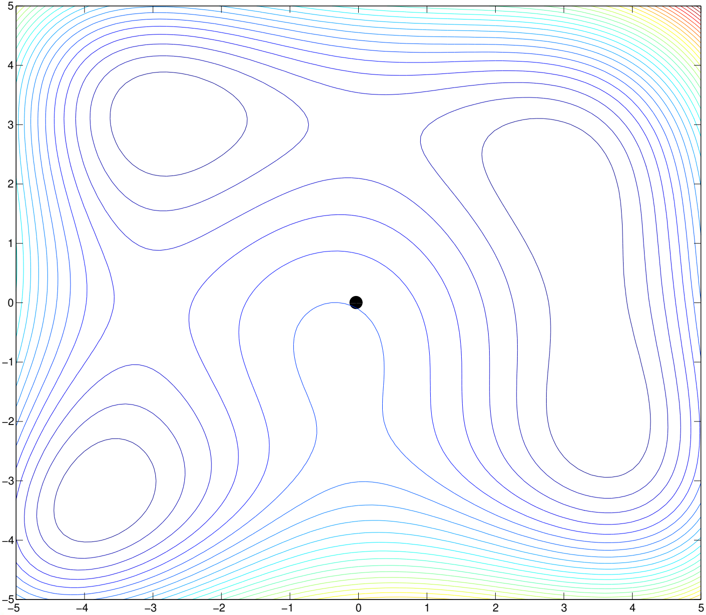
Numerical Optimization - Line Search



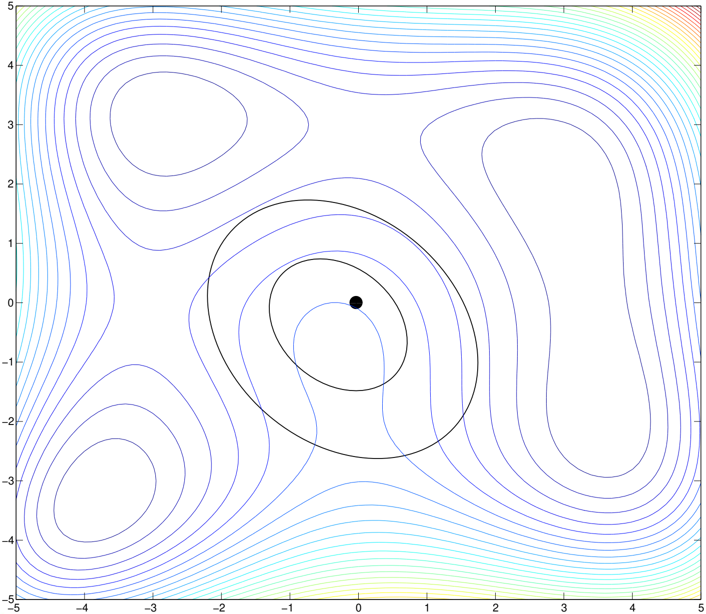
Trust Region Methods



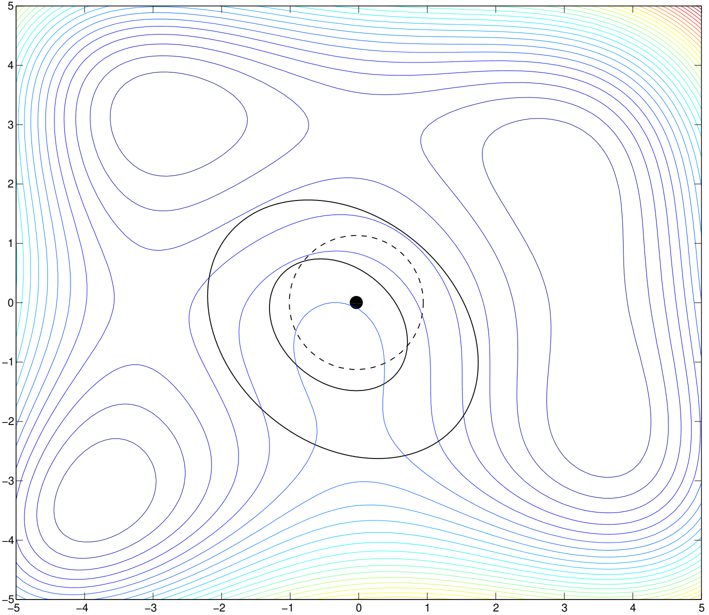
Trust Region Methods



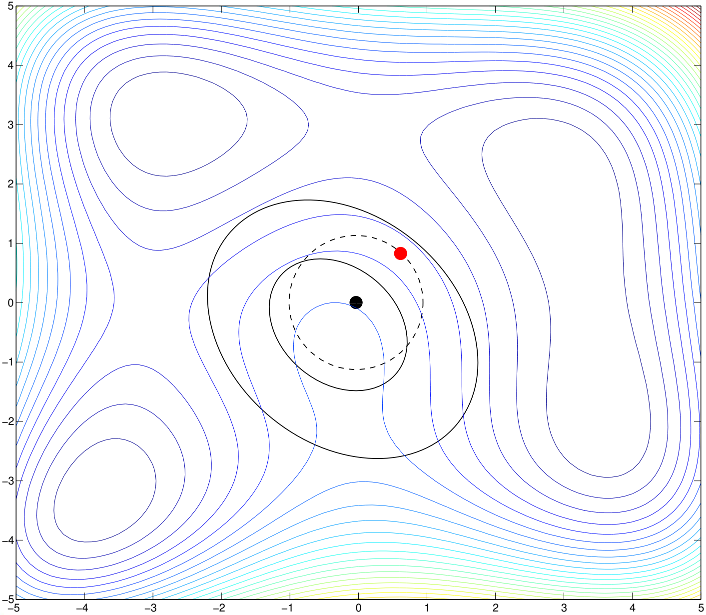
Trust Region Methods



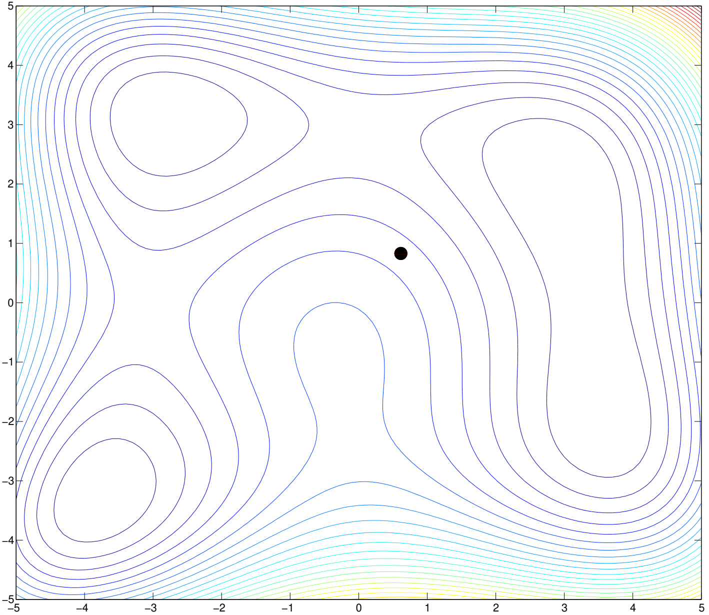
Trust Region Methods



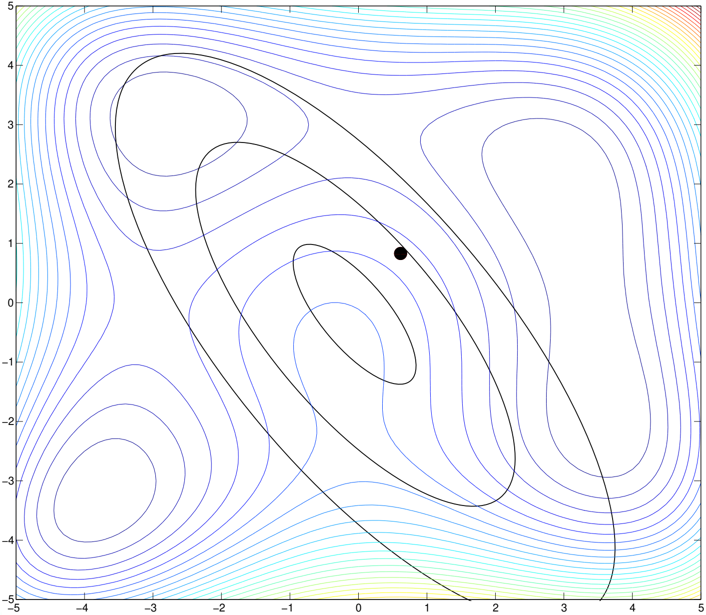
Trust Region Methods



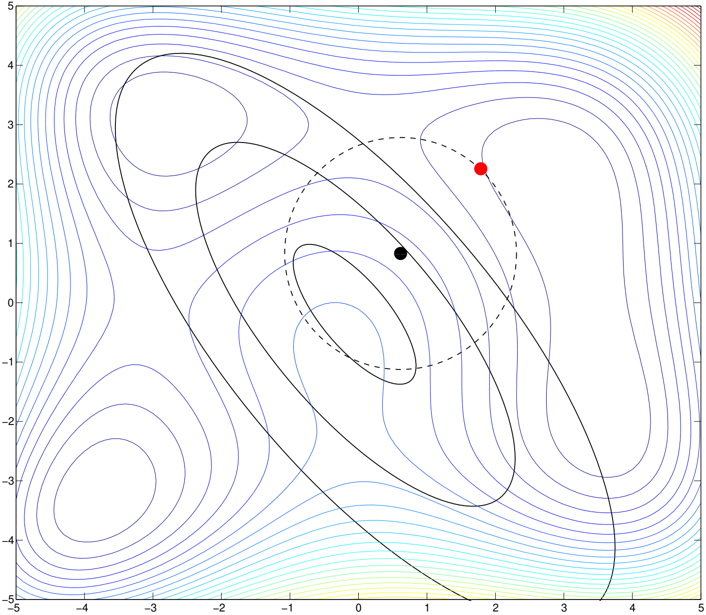
Trust Region Methods



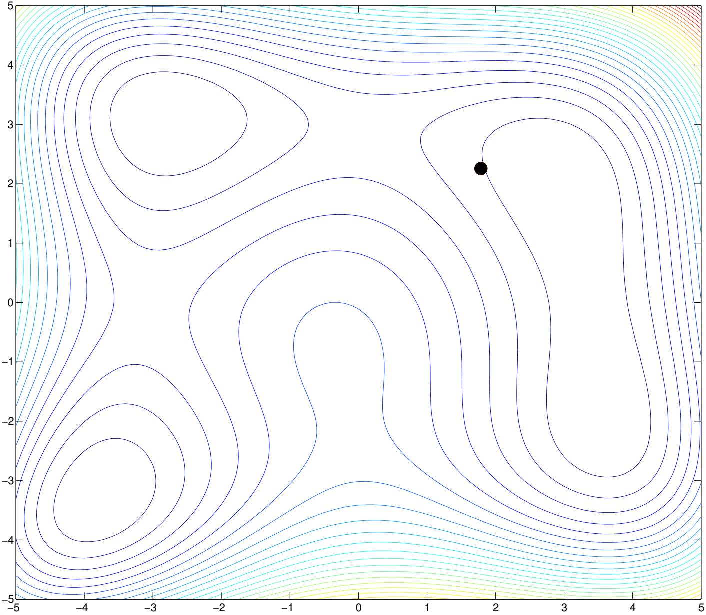
Trust Region Methods



Trust Region Methods



Trust Region Methods



Problem setup

$$\underset{x}{\text{minimize}}\ f(x; S(x))$$

$$\text{subject to: } x \in \mathcal{D} \subset \mathbb{R}^n$$

where the objective f depends on the output(s) from a simulation $S(x)$.

Problem setup

$$\underset{x}{\text{minimize}}\ f(x; S(x))$$

$$\text{subject to: } x \in \mathcal{D} \subset \mathbb{R}^n$$

where the objective f depends on the output(s) from a simulation $S(x)$.

- Derivatives of S may not be available
- Constraints defining $\mathcal{D}$ may or may not depend on S
- The dimension n is small
- Evaluating S is expensive
- f and/or S may be noisy. If the noise is stochastic,

$$\underset{x}{\text{minimize}}\ \mathbb{E}\left[\bar{f}(x)\right].$$

Initial approaches

- Grid over the domain (easily parallelizable)

Initial approaches

- Grid over the domain (easily parallelizable)
- Random sampling (easily parallelizable)

Initial approaches

- Grid over the domain (easily parallelizable)
- Random sampling (easily parallelizable)
- Evolutionary Algorithms (many are parallelizable)

Initial approaches

- Grid over the domain (easily parallelizable)
- Random sampling (easily parallelizable)
- Evolutionary Algorithms (many are parallelizable)
  - Genetic Algorithm
  - Simulated Annealing
  - Particle Swarm
  - Ant Colony Optimization
  - Bee Colony Optimization
  - Grey Wolf Optimization

DFO warnings

- Be careful
  1) A problem can be written as a scalar output, black box
  2) An algorithm exists to optimize a scalar output, black box function

  1) and 2) true doesn't mean the algorithm should be used

DFO warnings

- Be careful
  1) A problem can be written as a scalar output, black box
  2) An algorithm exists to optimize a scalar output, black box function

  1) and 2) true doesn't mean the algorithm should be used

$$\underset{x}{\text{minimize}}\, f(x) = \|Ax - b\|$$

DFO warnings

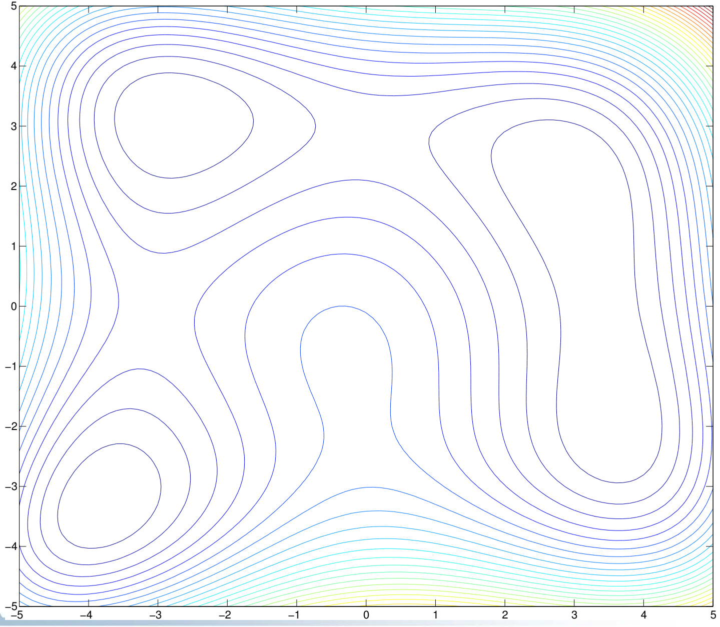
- Be careful
  1) A problem can be written as a scalar output, black box
  2) An algorithm exists to optimize a scalar output, black box function

  1) and 2) true doesn't mean the algorithm should be used

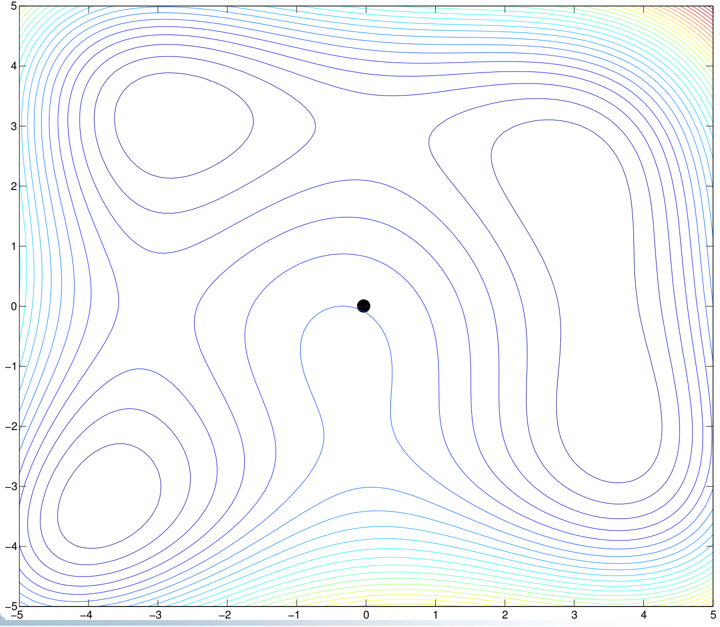
$$\underset{x}{\text{minimize}}\, f(x) = \|Ax - b\|$$

- If your problem has derivatives, please use them. If you don't have them...
  - Algorithmic Differentiation (AD) is wonderful
- Does the problem have structure? Avoid black boxes

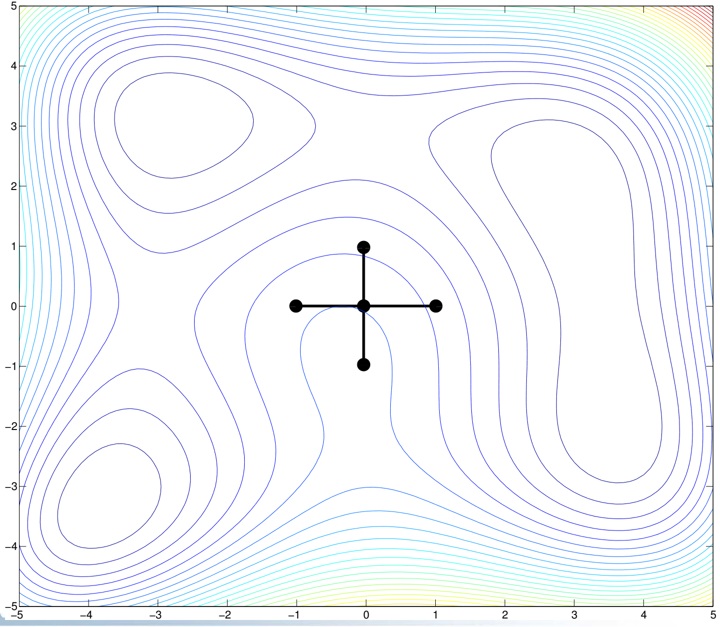
Coordinate Search



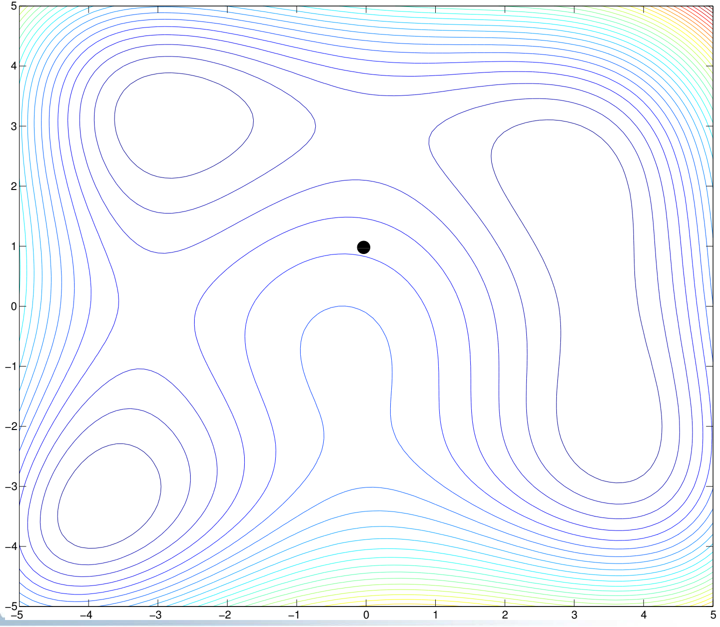
Coordinate Search



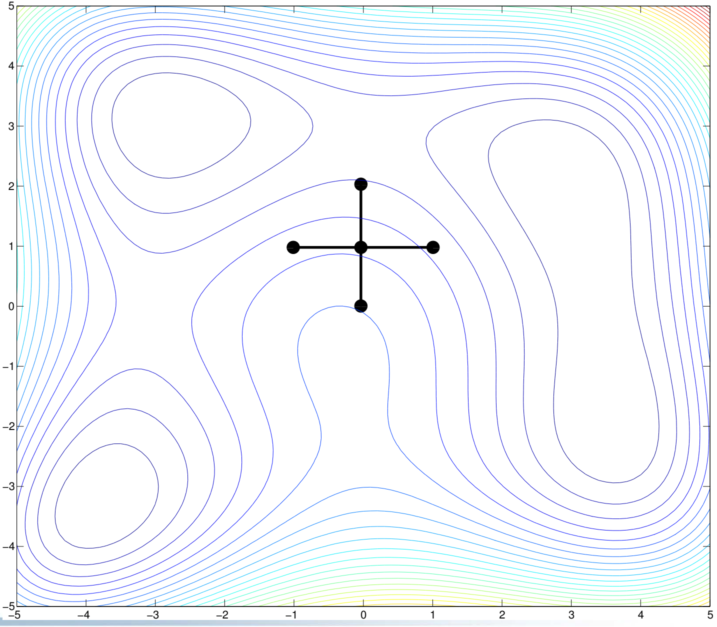
Coordinate Search



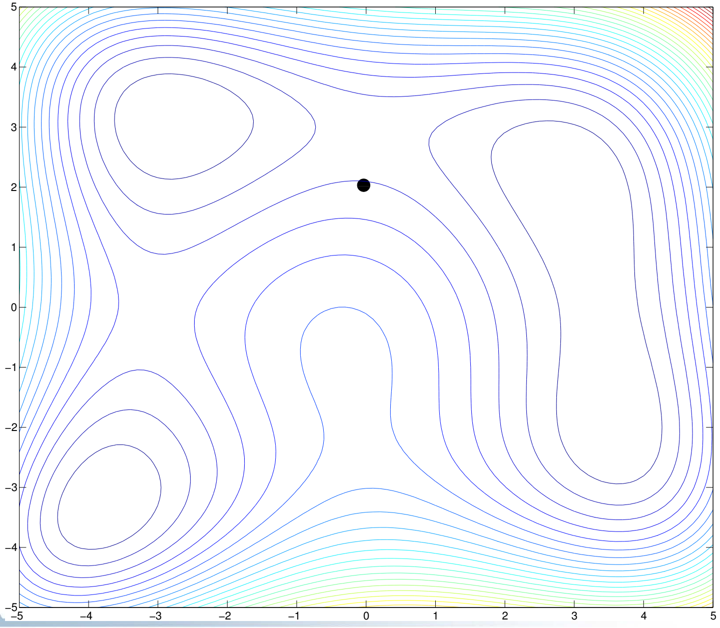
Coordinate Search



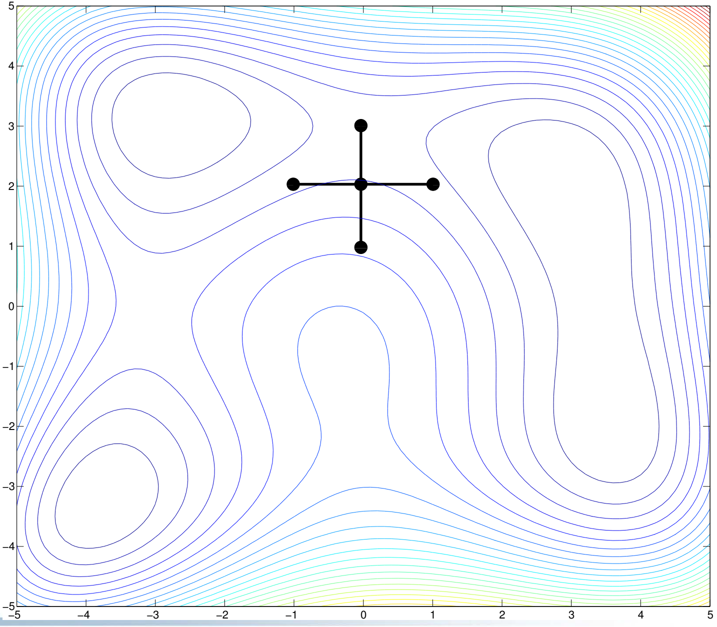
Coordinate Search



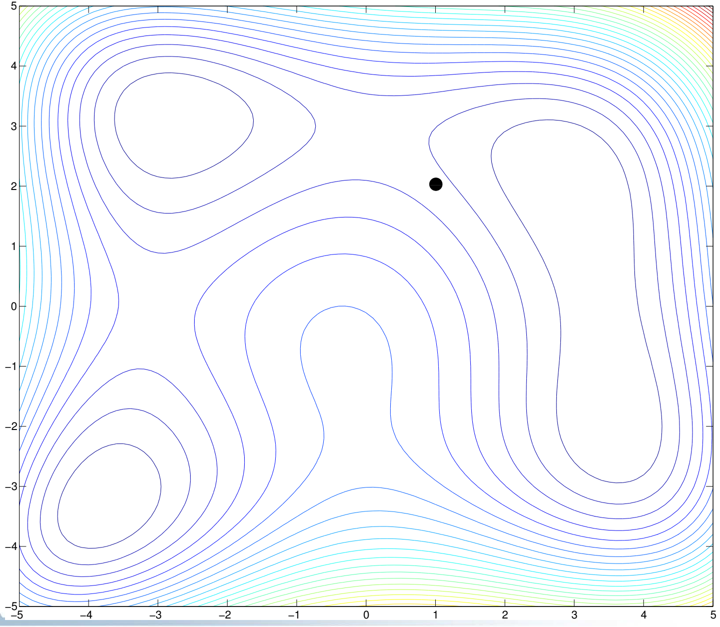
Coordinate Search



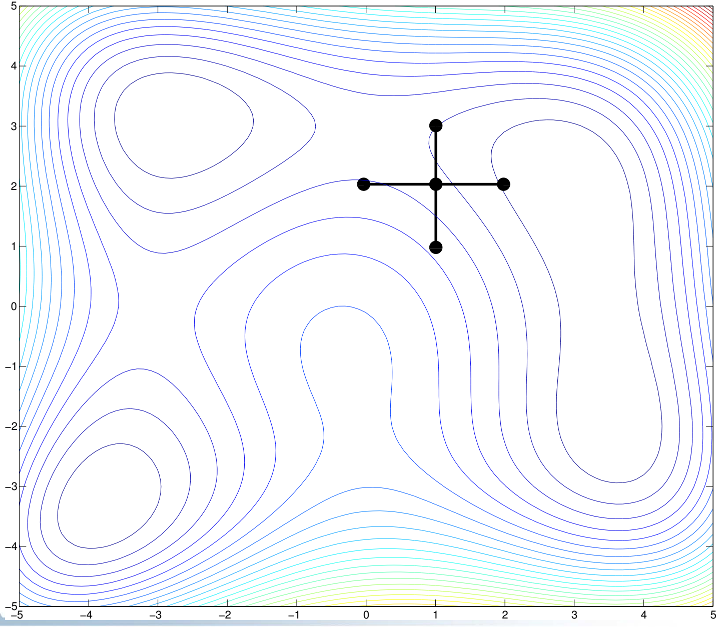
Coordinate Search



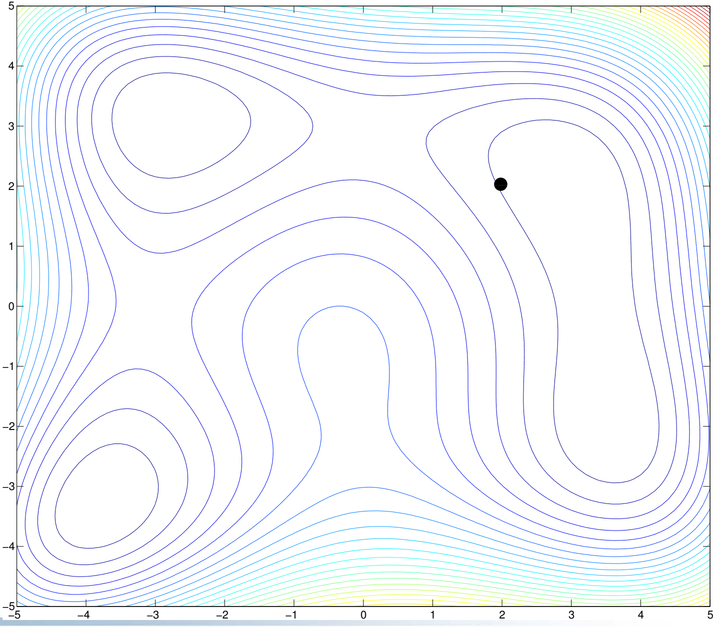
Coordinate Search



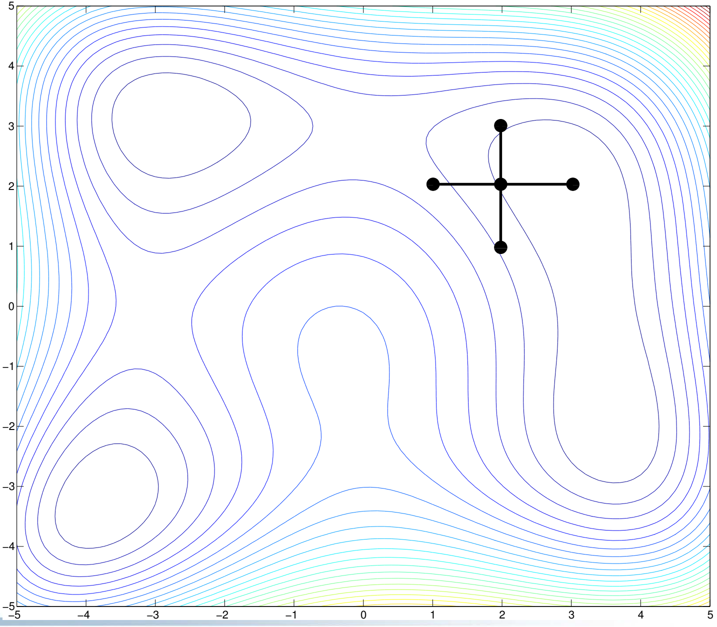
Coordinate Search



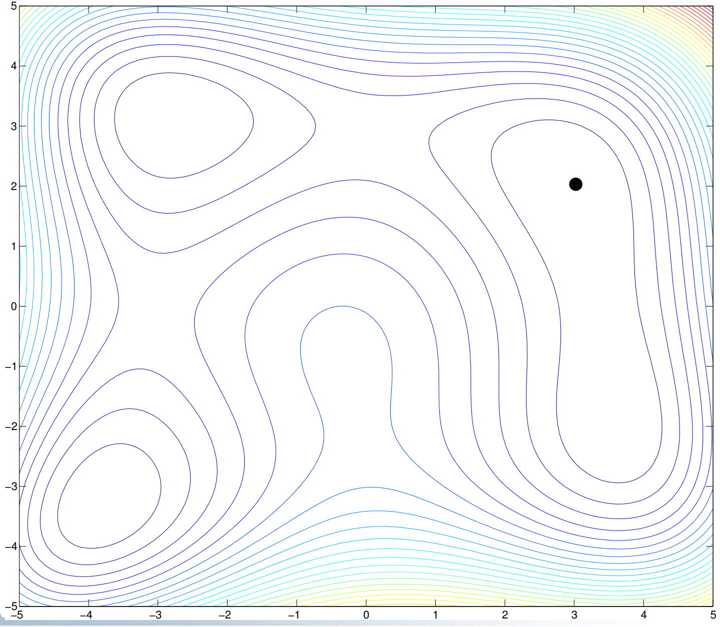
Coordinate Search



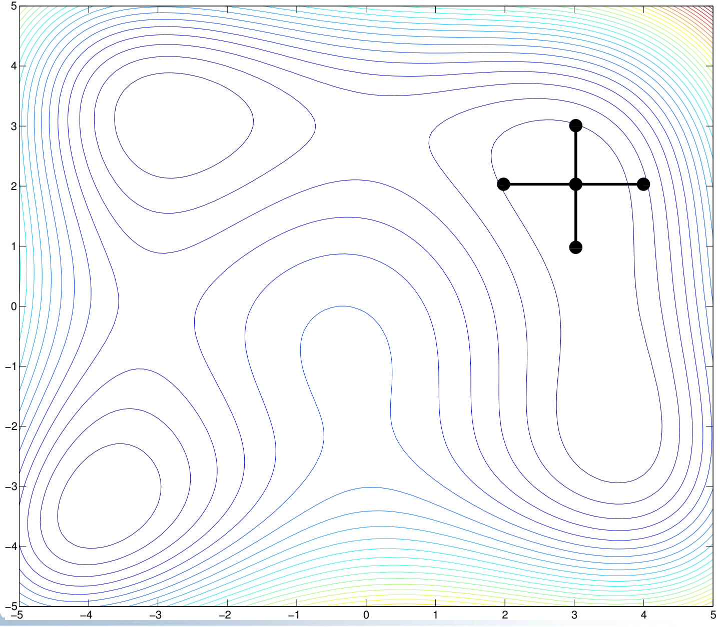
Coordinate Search



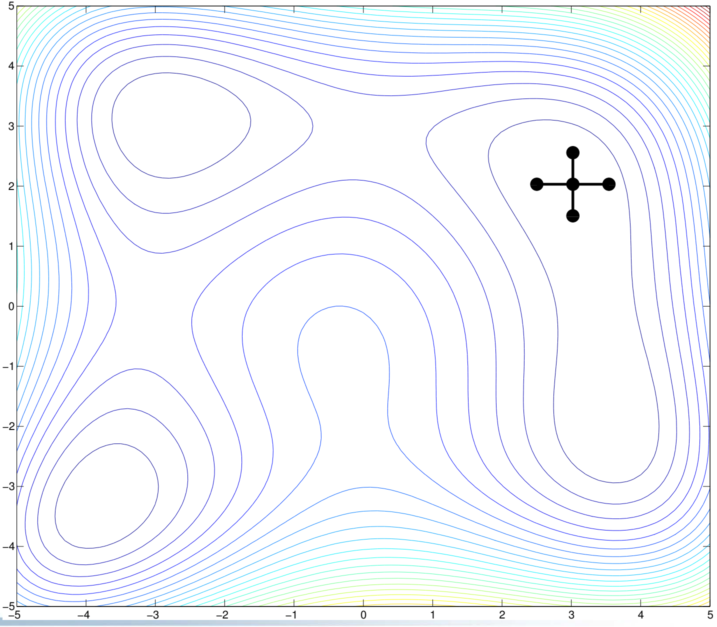
Coordinate Search



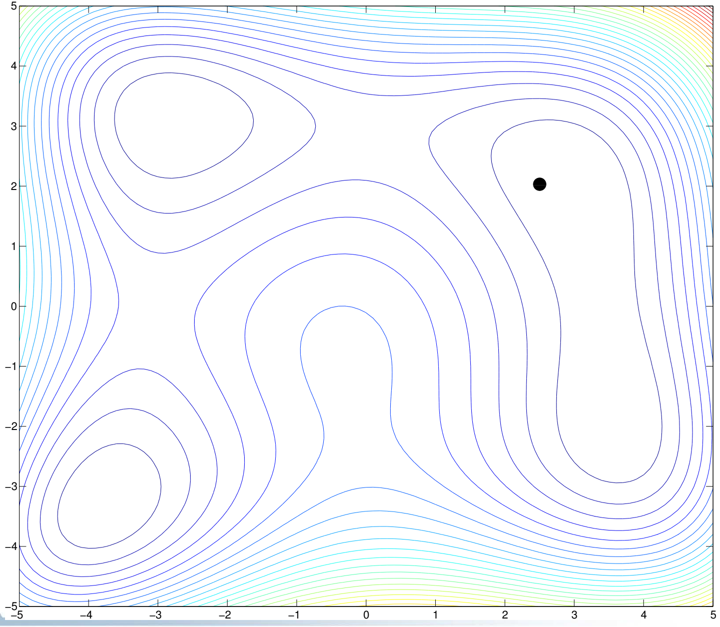
Coordinate Search



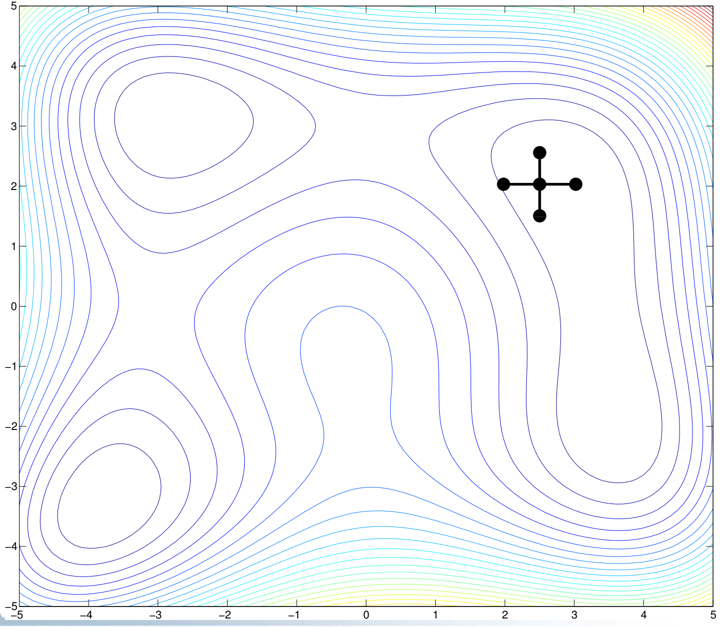
Coordinate Search



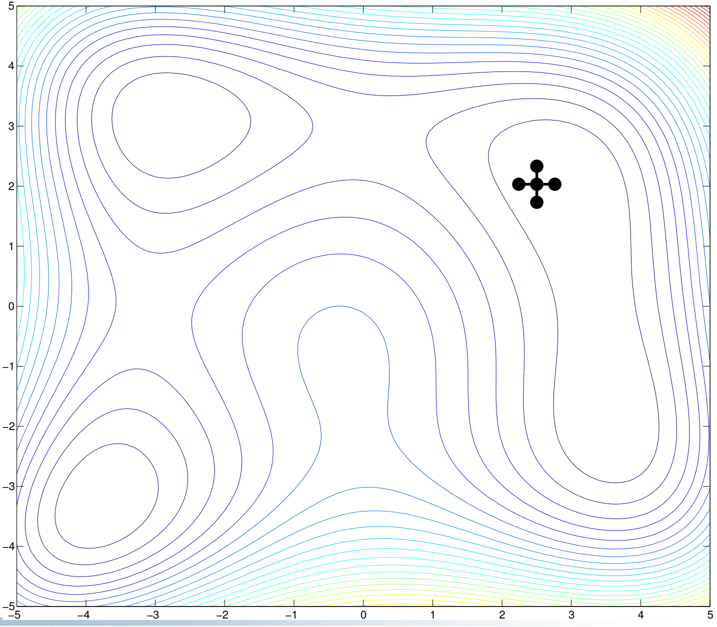
Coordinate Search



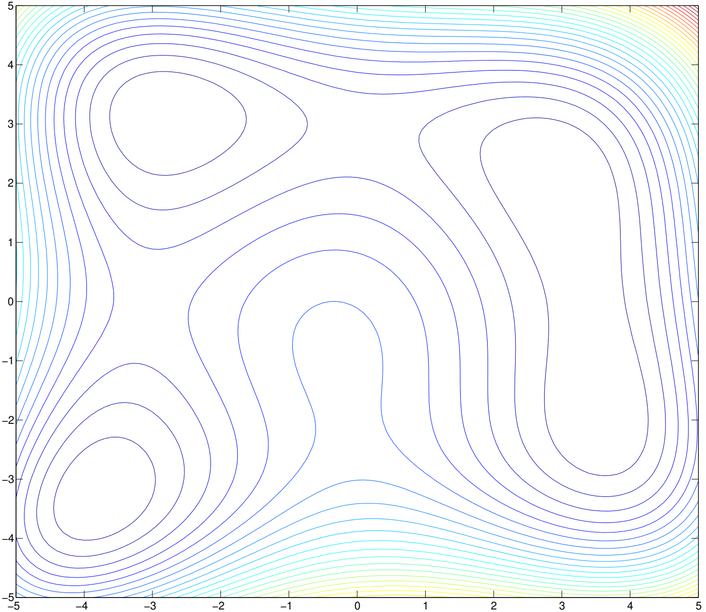
Coordinate Search



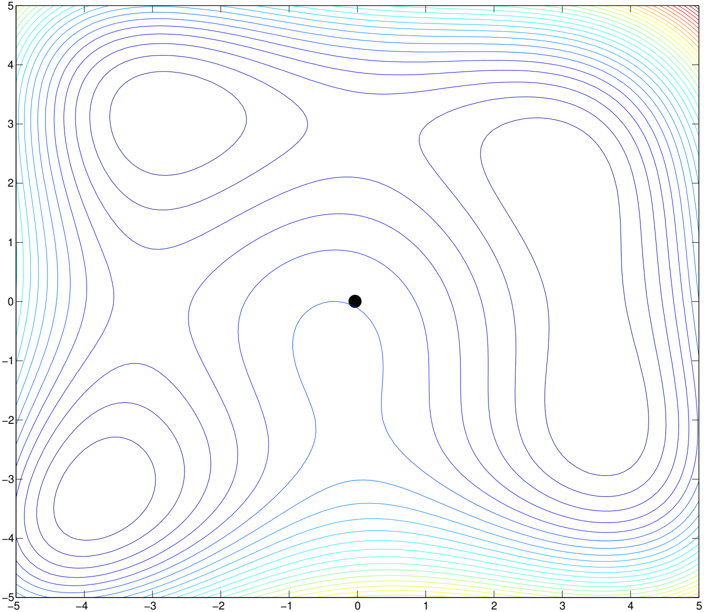
Coordinate Search



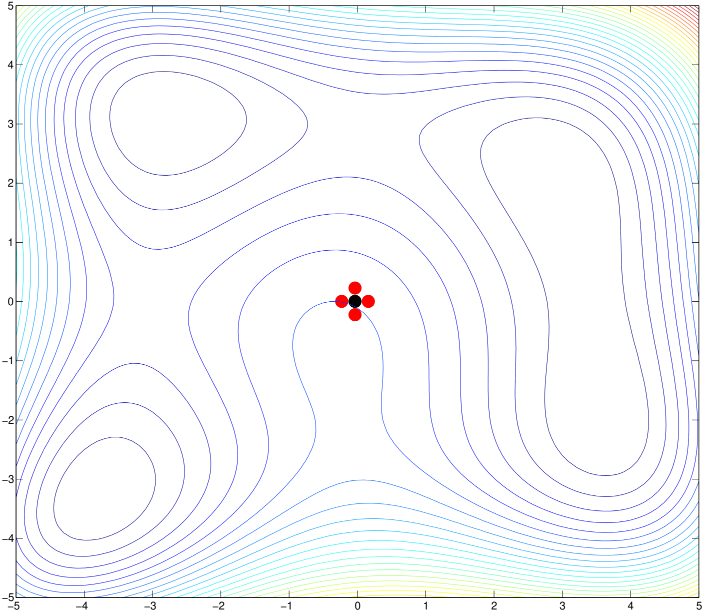
Approximate Gradients



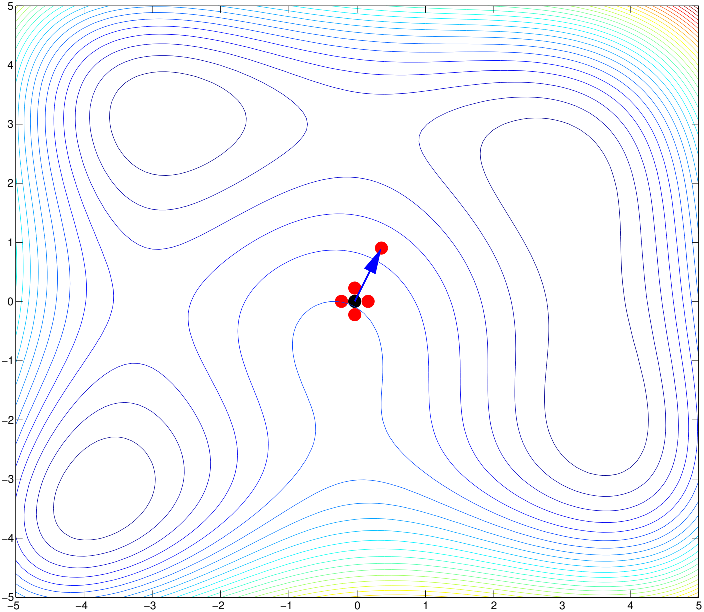
Approximate Gradients



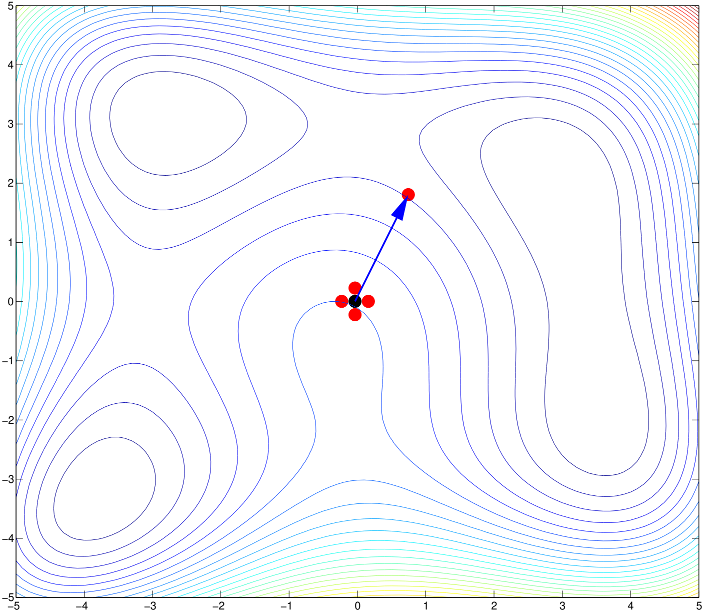
Approximate Gradients



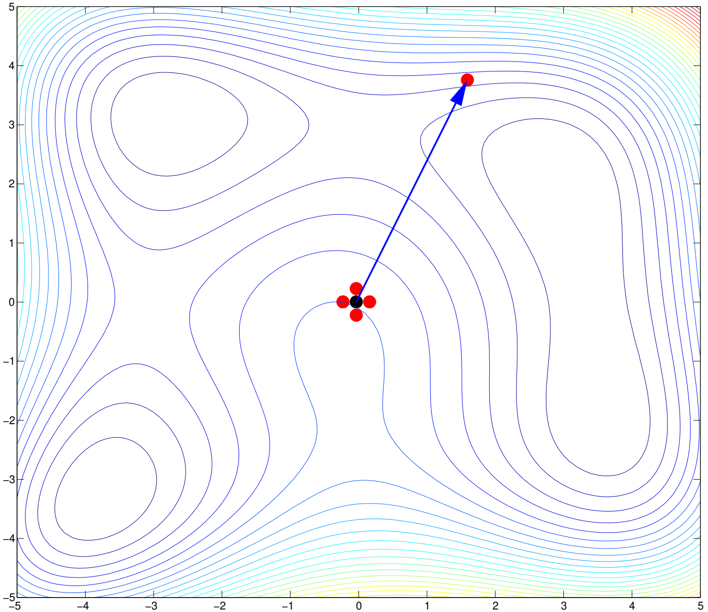
Approximate Gradients



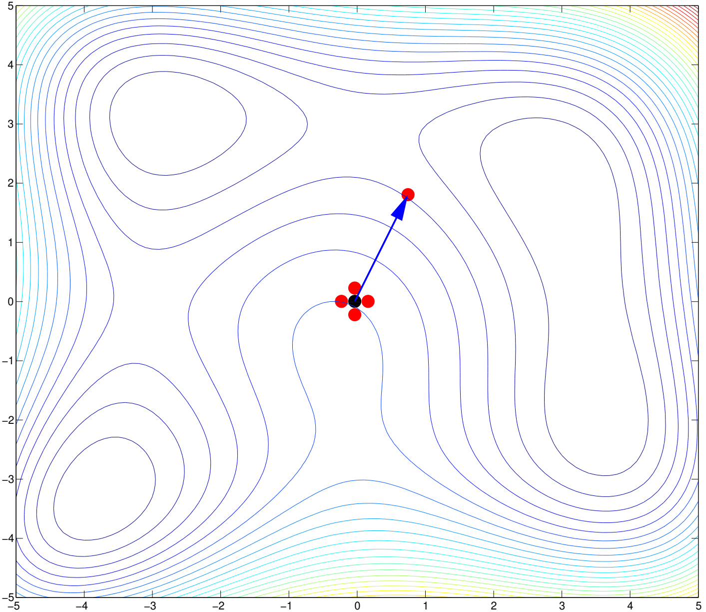
Approximate Gradients



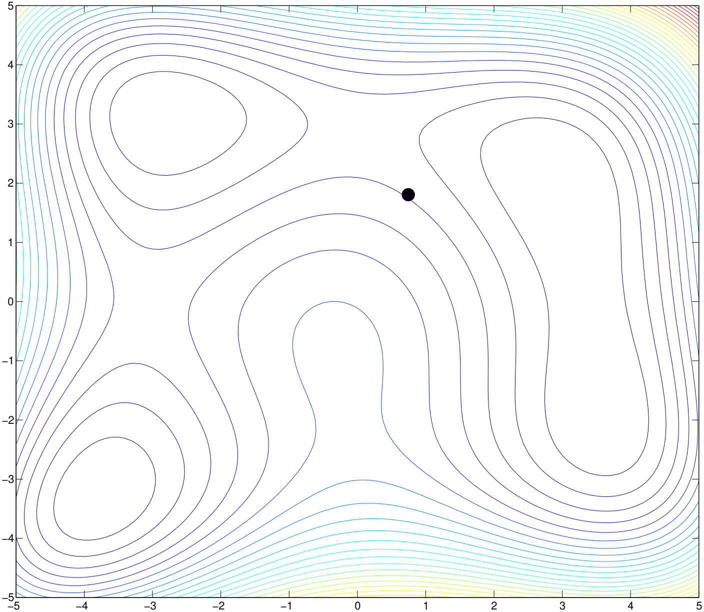
Approximate Gradients



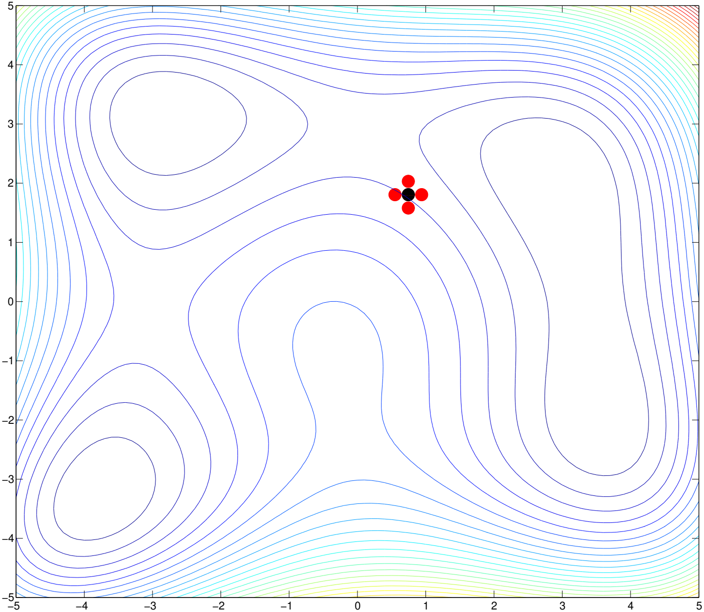
Approximate Gradients



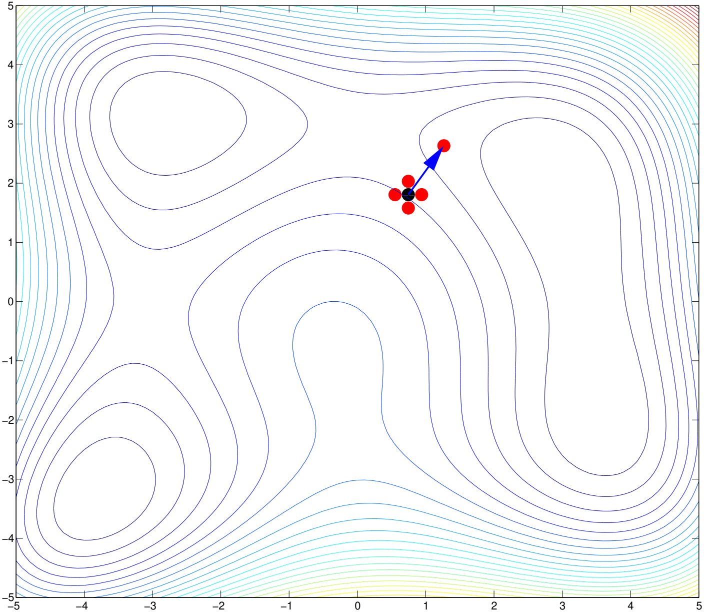
Approximate Gradients



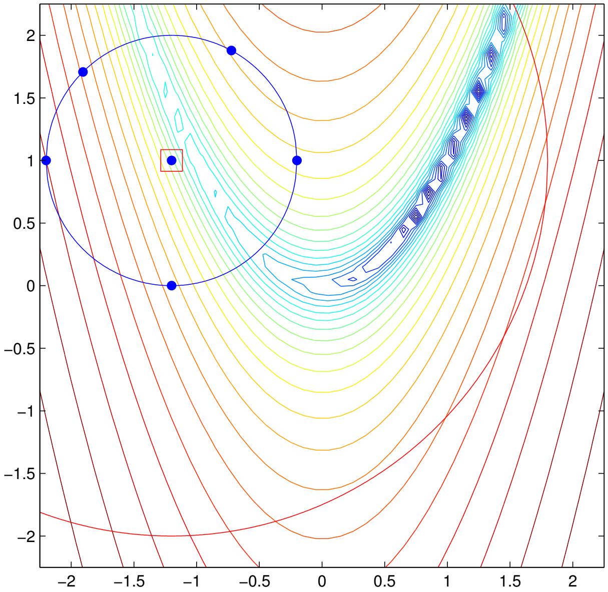
Approximate Gradients



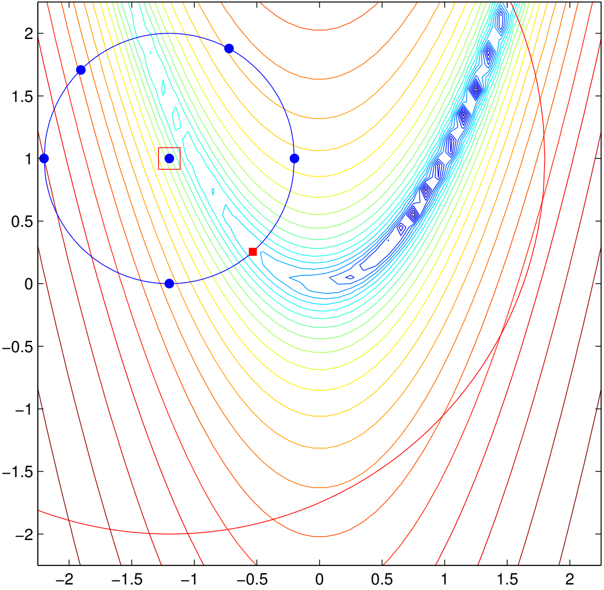
Approximate Gradients



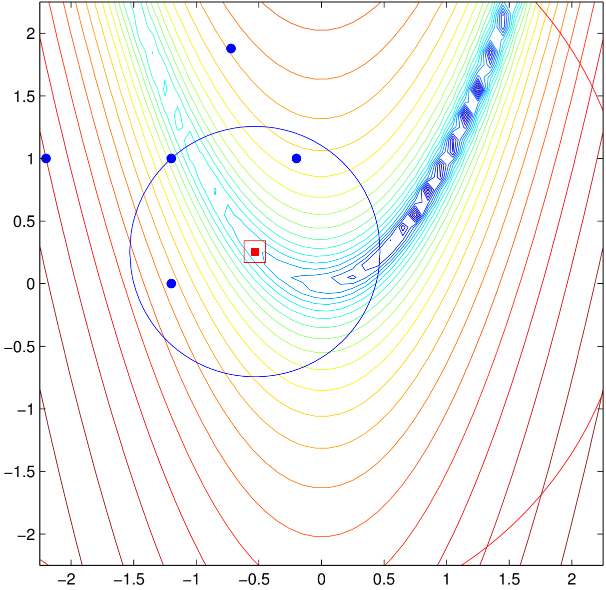
Model-based methods - Interpolation



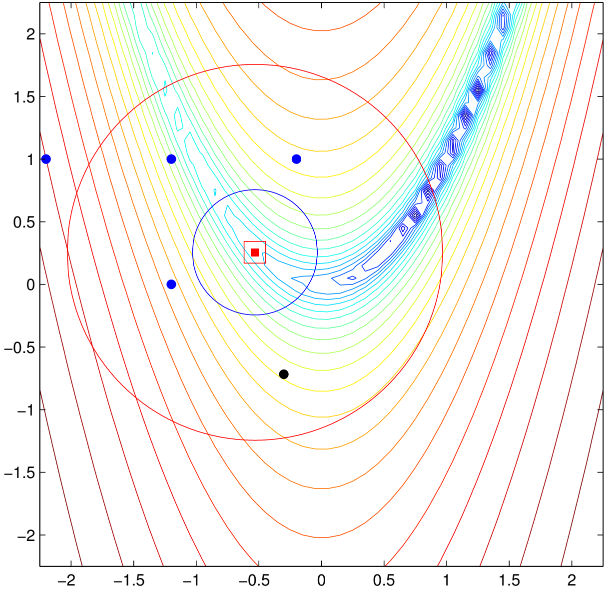
Model-based methods - Interpolation



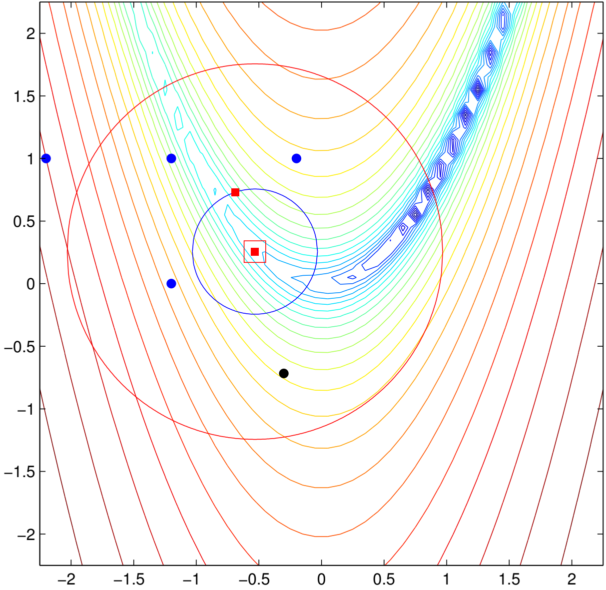
Model-based methods - Interpolation



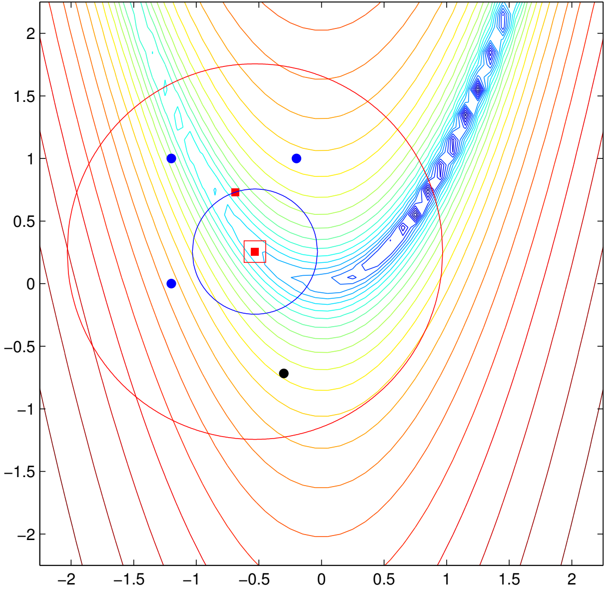
Model-based methods - Interpolation



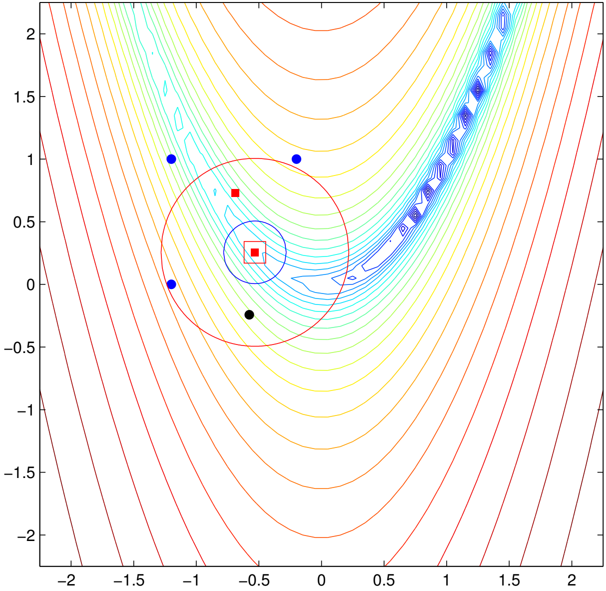
Model-based methods - Interpolation



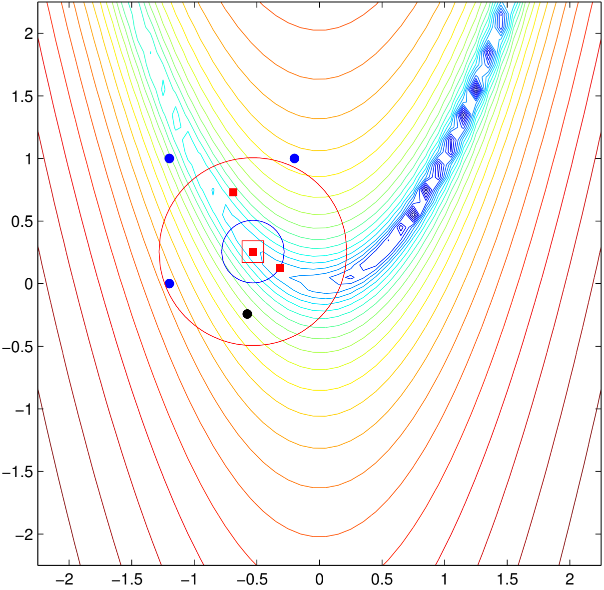
Model-based methods - Interpolation



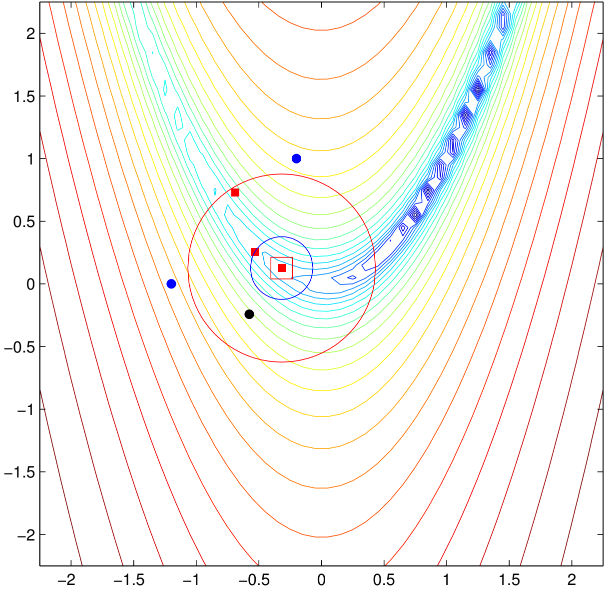
Model-based methods - Interpolation



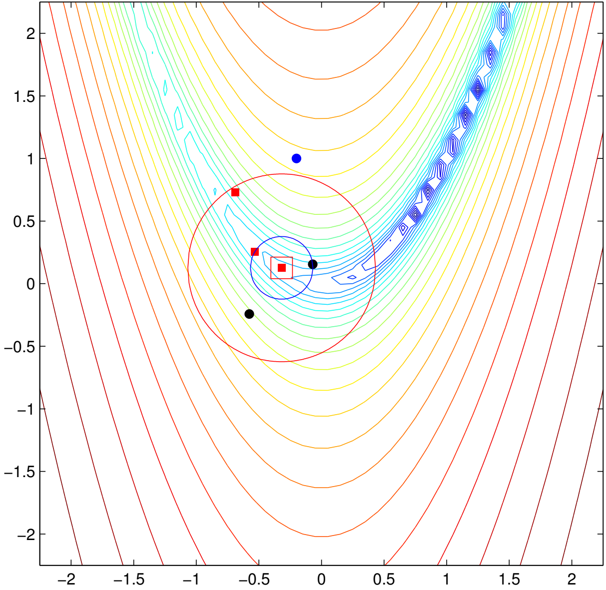
Model-based methods - Interpolation



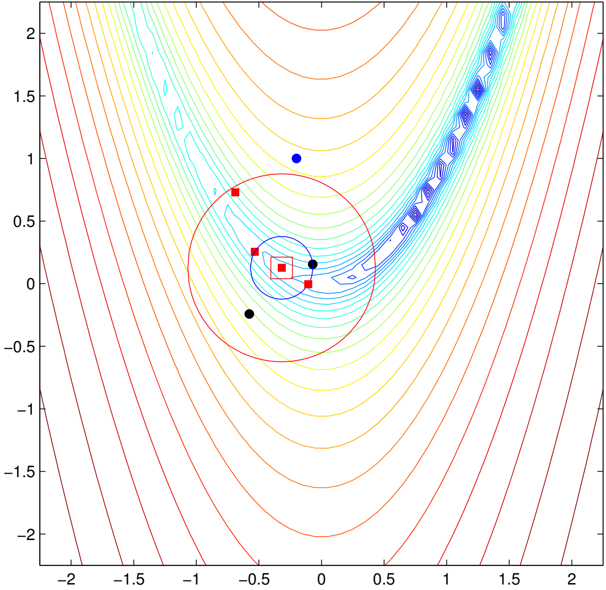
Model-based methods - Interpolation



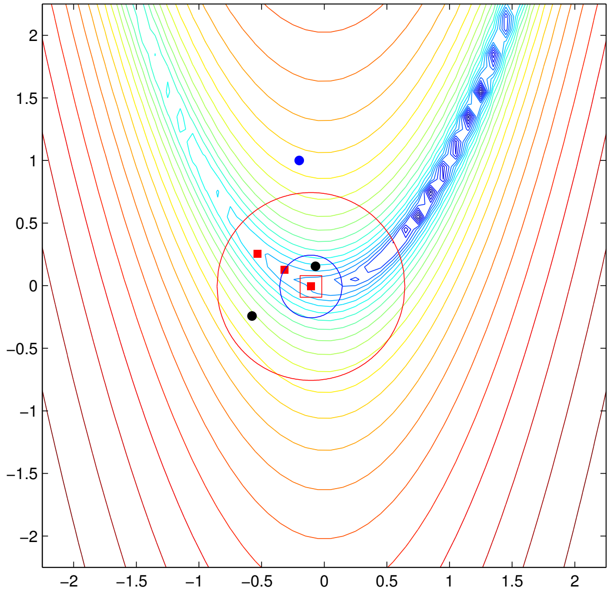
Model-based methods - Interpolation



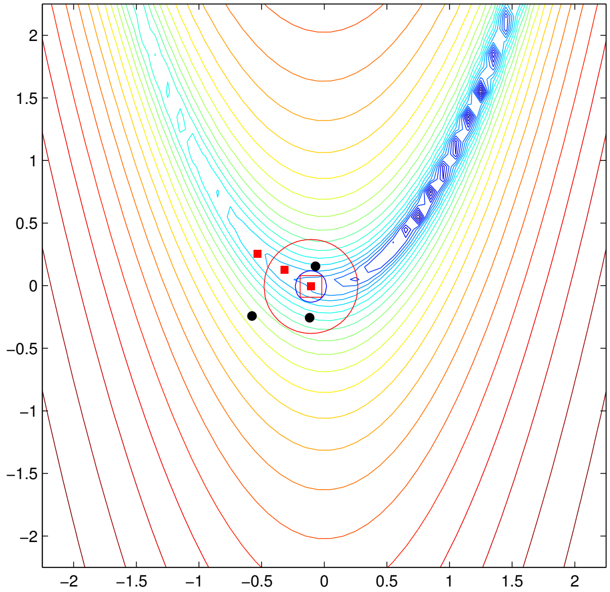
Model-based methods - Interpolation



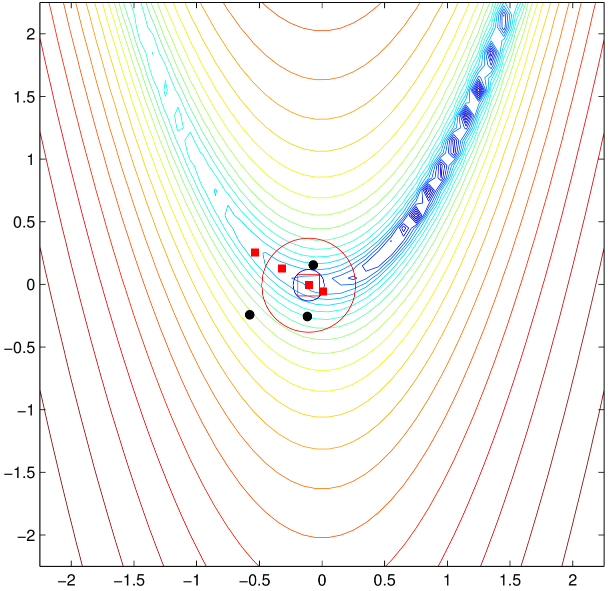
Model-based methods - Interpolation



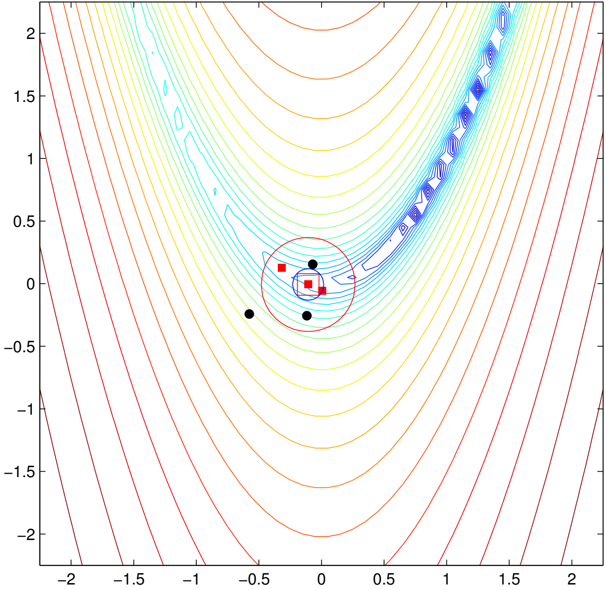
Model-based methods - Interpolation



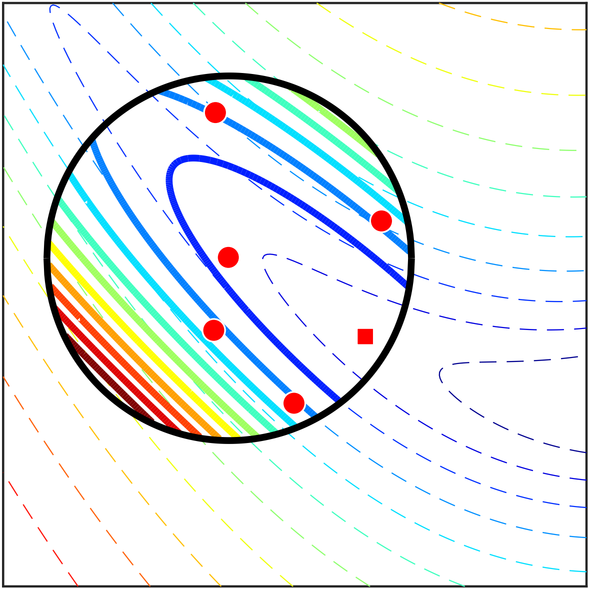
Model-based methods - Interpolation



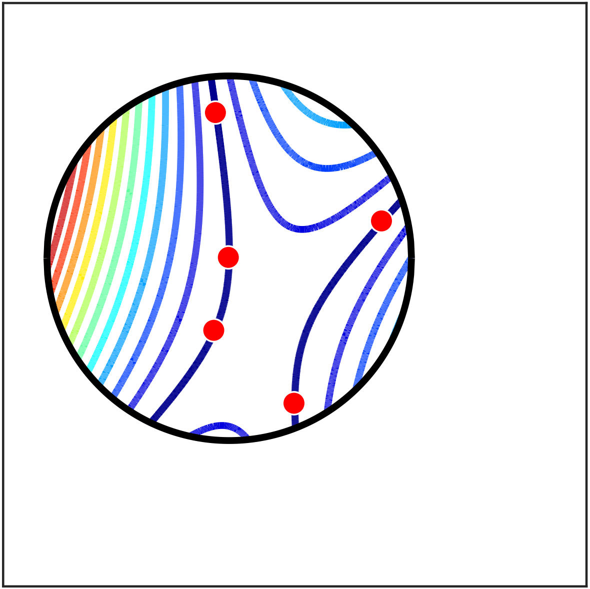
Model-based methods - Interpolation



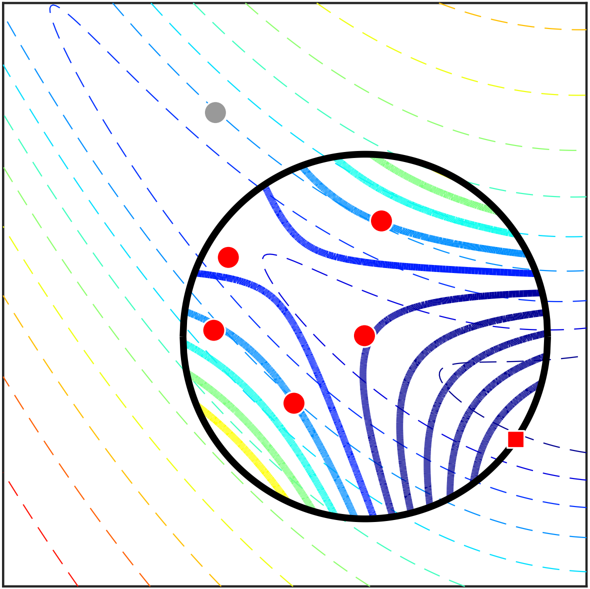
Model-based methods - Interpolation



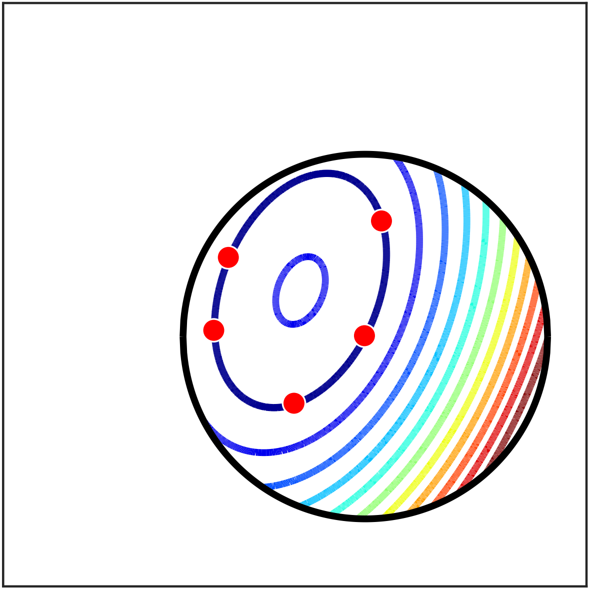
Model-based methods - Interpolation



Model-based methods - Interpolation



Model-based methods - Interpolation



DFO warnings

- Be careful
  1) A problem can be written as a scalar output, black box
  2) An algorithm exists to optimize a scalar output, black box function

  1) and 2) true doesn't mean the algorithm should be used

$$\underset{x}{\text{minimize}}\, f(x) = \|Ax - b\|$$

- If your problem has derivatives, please use them. If you don't have them...
  - Algorithmic Differentiation (AD) is wonderful

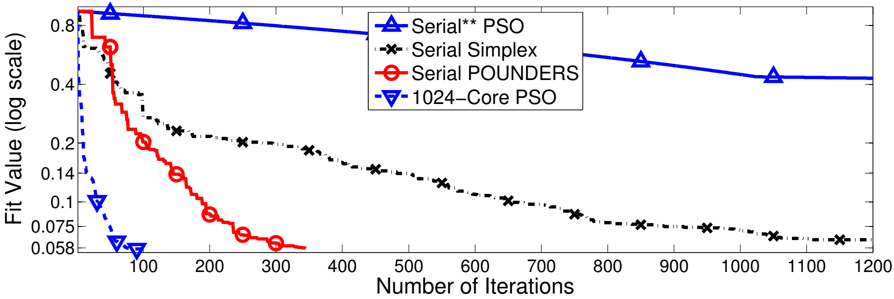
- Does the problem have structure? Avoid black boxes

Opening up the black box

$$f(x) = \sum_{i=1}^{r} (F_i(x) - T_i)^2$$

Can either have a solver that uses $f(x)$ or $[F_1(x), \ldots, F_r(x)]$.

Opening up the black box

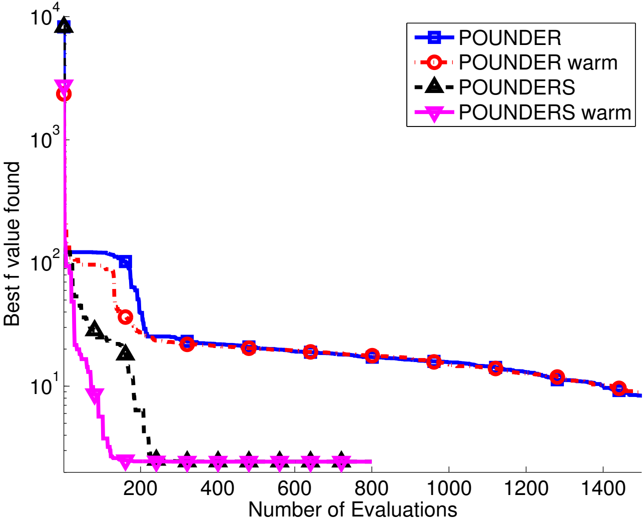


Tuning quadrupole moments for a particle accelerator simulation.

$$f(x) = \sum_{i=1}^{r} (F_i(x) - T_i)^2$$

Can either have a solver that uses $f(x)$ or $[F_1(x), \ldots, F_r(x)]$.

Opening up the black box

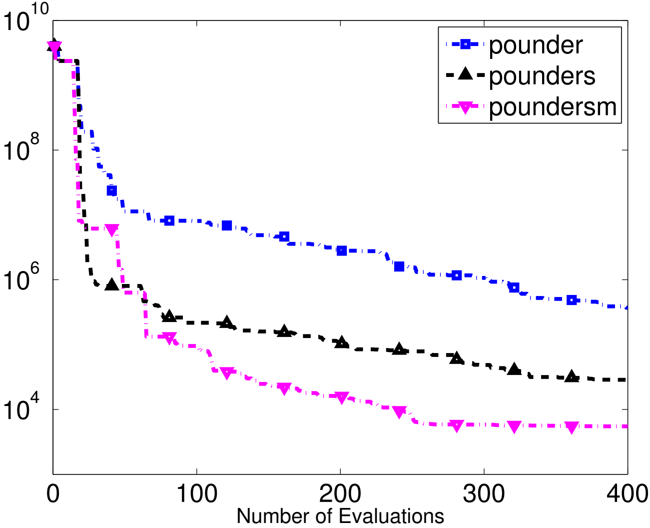


$F(x) : \mathbb{R}^{16} \to \mathbb{R}^{2049}$

$2n$ **experimental design around starting point**

Energy density functional calibrations.

Opening up the black box

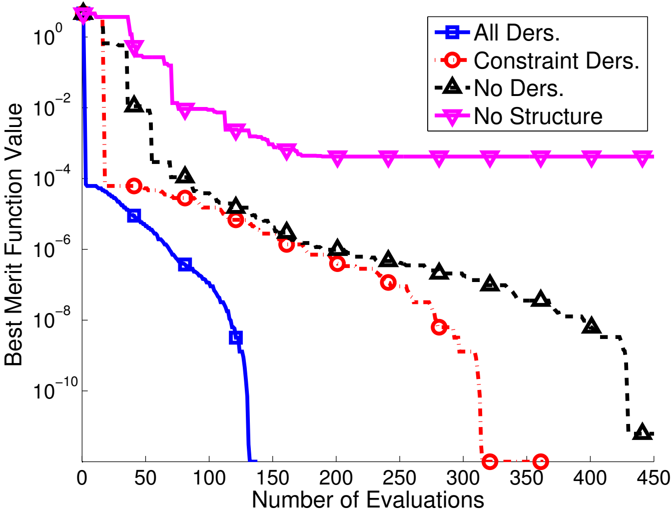


Energy density functional calibrations.

$F(x) : \mathbb{R}^{16} \to \mathbb{R}^{2049}$

$F_i(x) = B([x]_{1-13}) + g(x)$

Opening up the black box



Small gas network problem.

- 15 variables
- 11 constraints

- $\nabla_x f$ and $\nabla_x c$
- f and $\nabla_x c$
- f and c (separate) black boxes
- Penalizing constraints

Exploiting Structure

- Nonsmooth, composite optimization

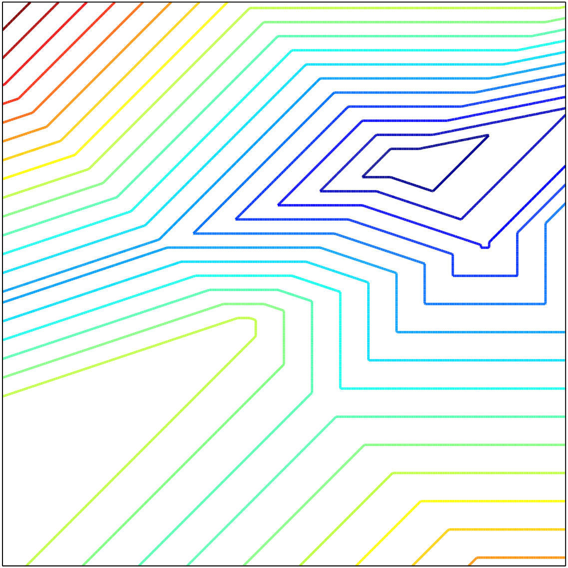
$$\underset{x}{\text{minimize}}\, f(x) = h(F(x))$$

where ∇F is unavailable but ∂h is known

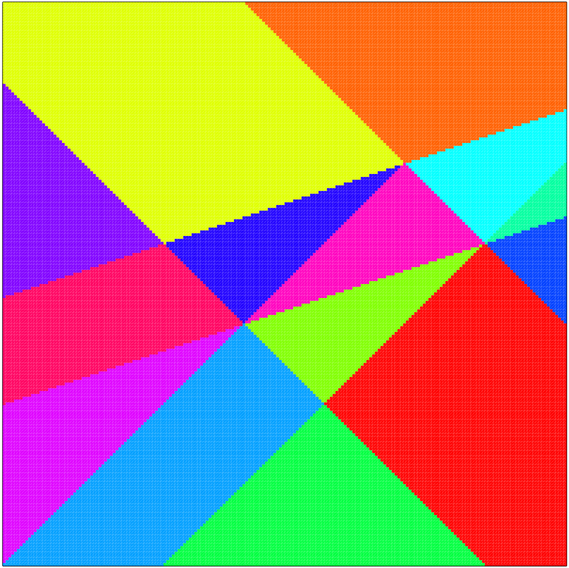
Example

$$f(x) = \sum_{i=1}^{r} |d_i - \max\{c_i, F_i(x)\}|$$

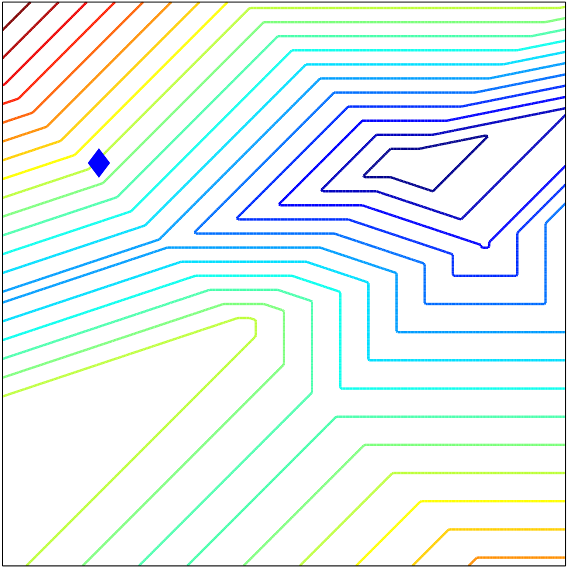
Nonsmooth, composite optimization



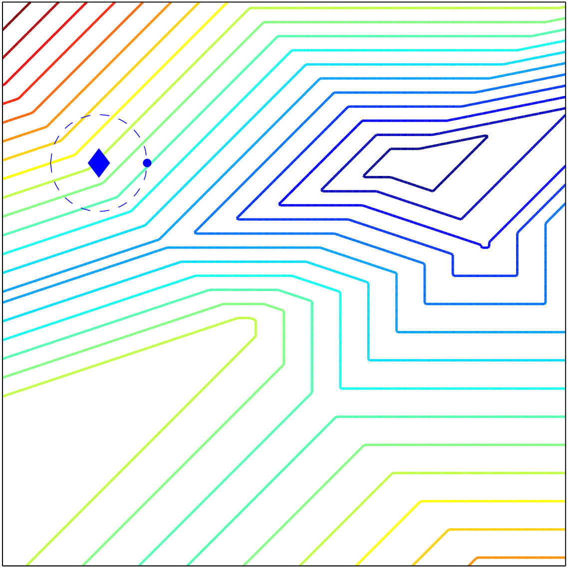
Nonsmooth, composite optimization



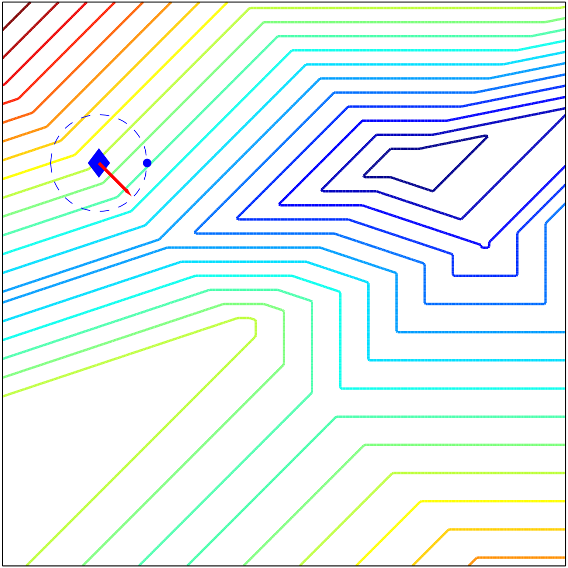
Nonsmooth, composite optimization



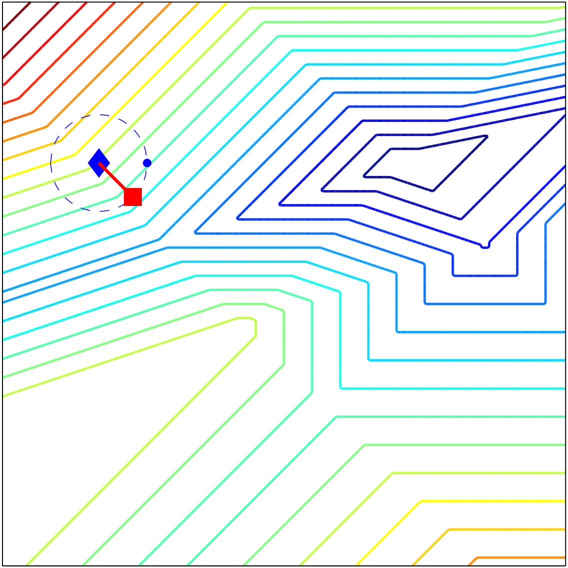
Nonsmooth, composite optimization



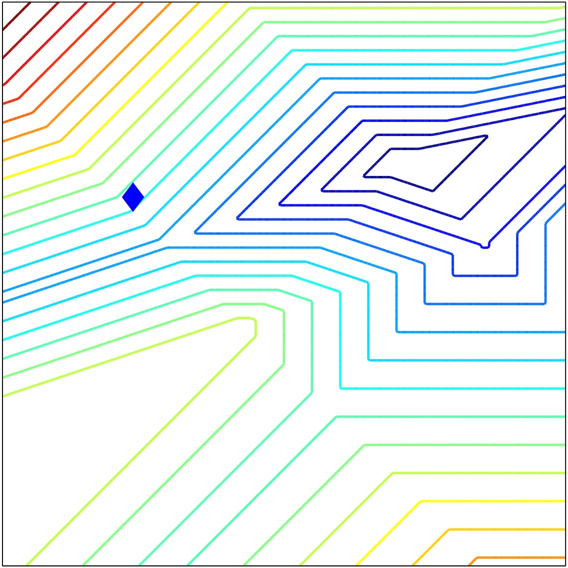
Nonsmooth, composite optimization



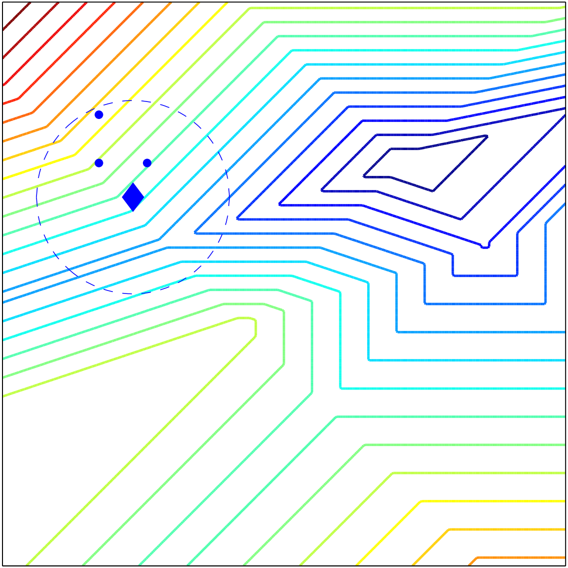
Nonsmooth, composite optimization



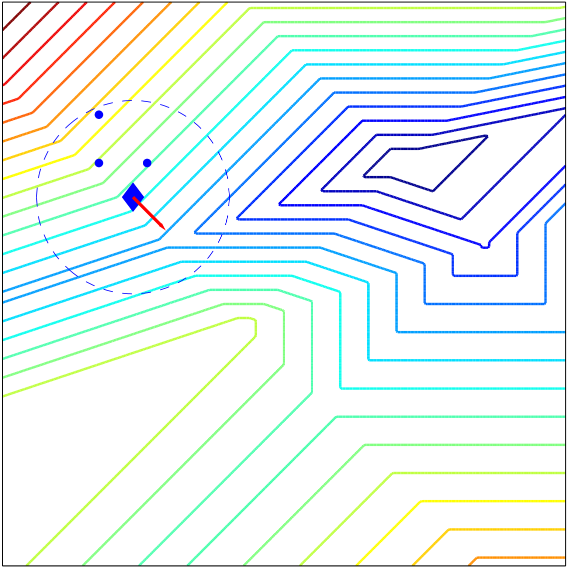
Nonsmooth, composite optimization



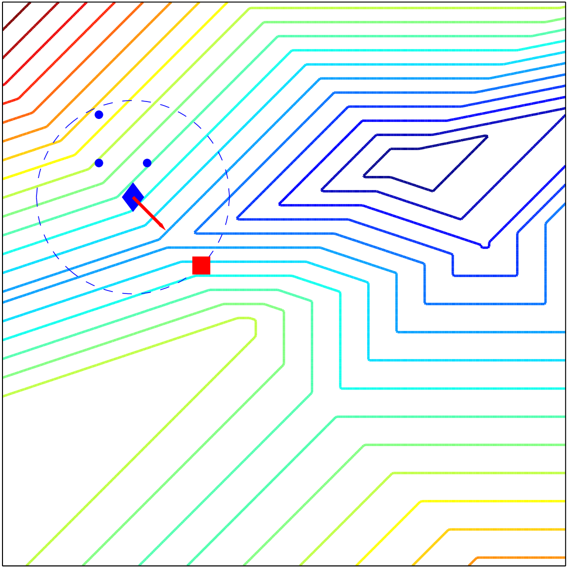
Nonsmooth, composite optimization



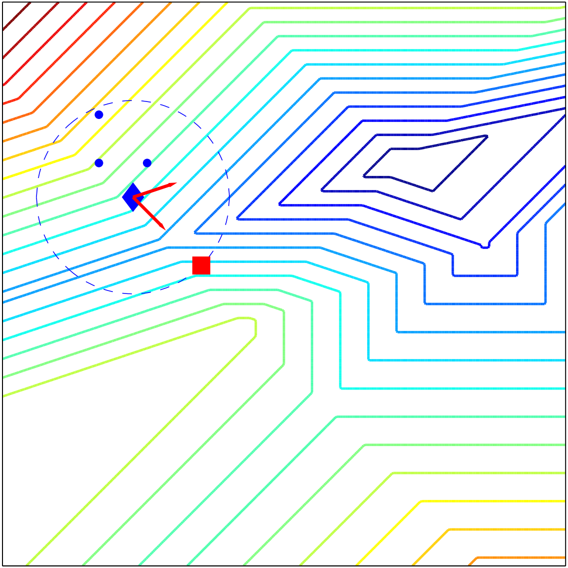
Nonsmooth, composite optimization



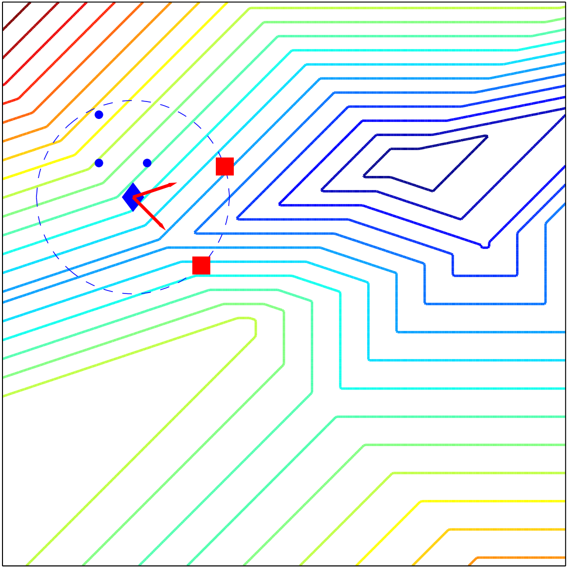
Nonsmooth, composite optimization



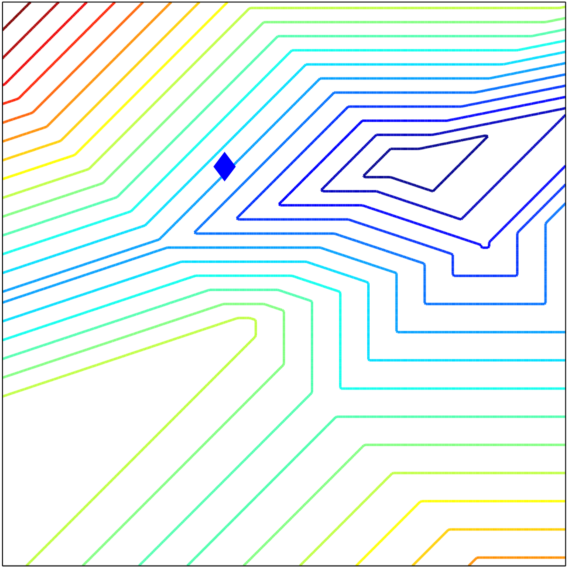
Nonsmooth, composite optimization



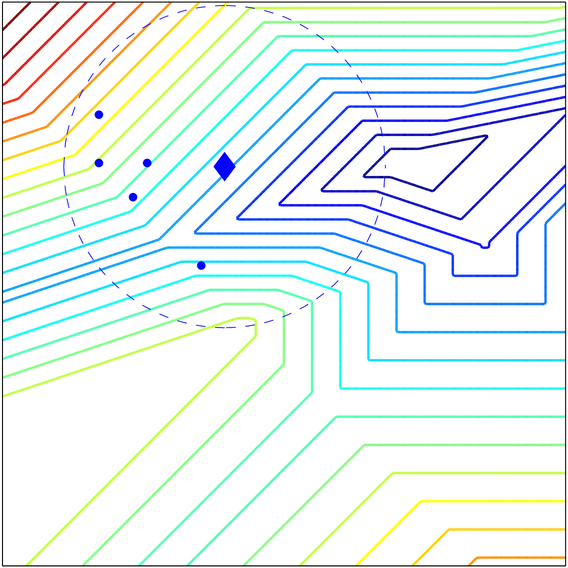
Nonsmooth, composite optimization



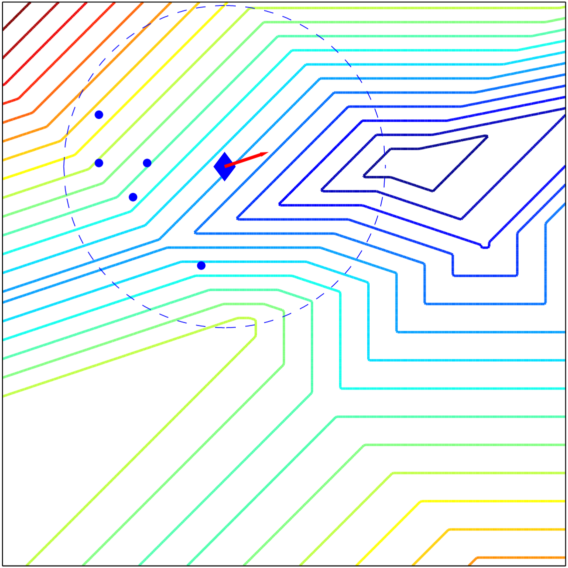
Nonsmooth, composite optimization



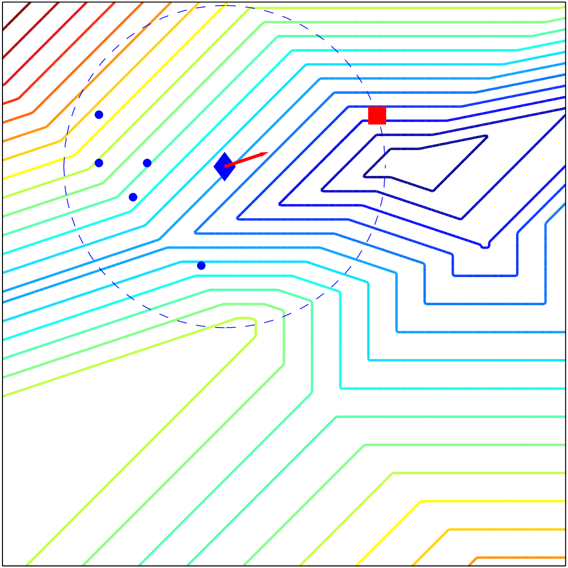
Nonsmooth, composite optimization



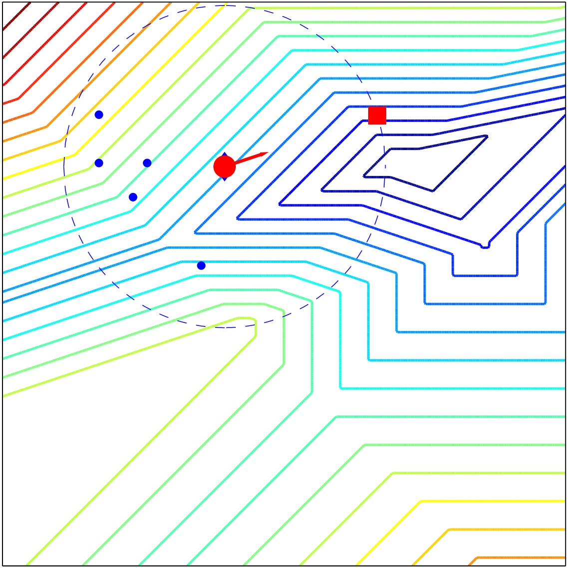
Nonsmooth, composite optimization



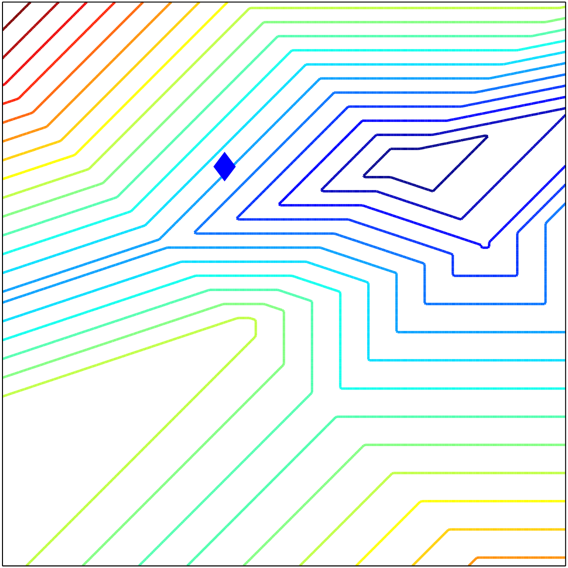
Nonsmooth, composite optimization



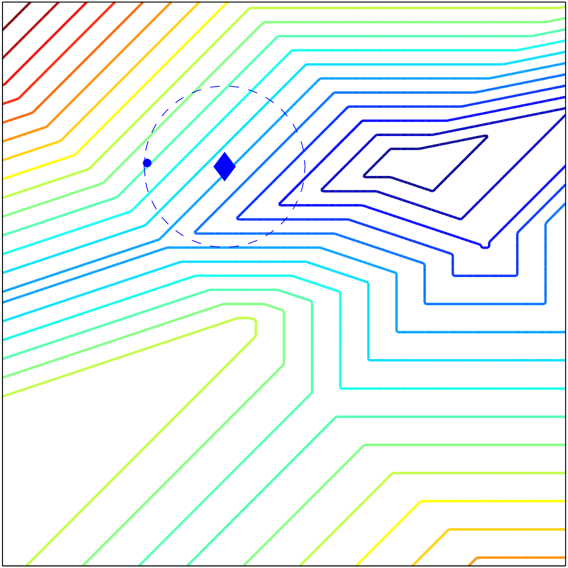
Nonsmooth, composite optimization



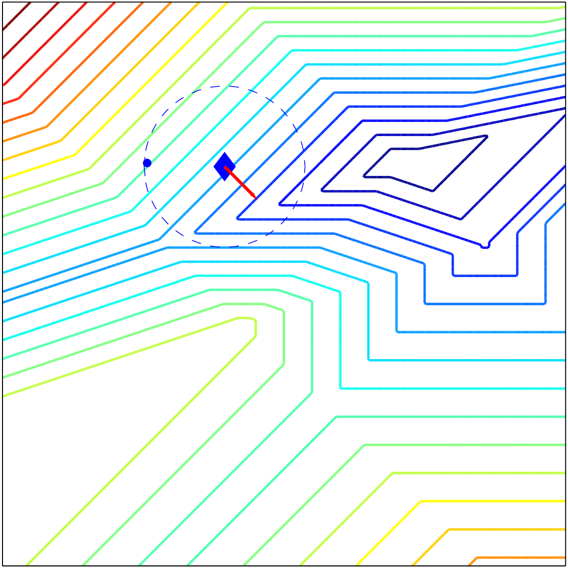
Nonsmooth, composite optimization



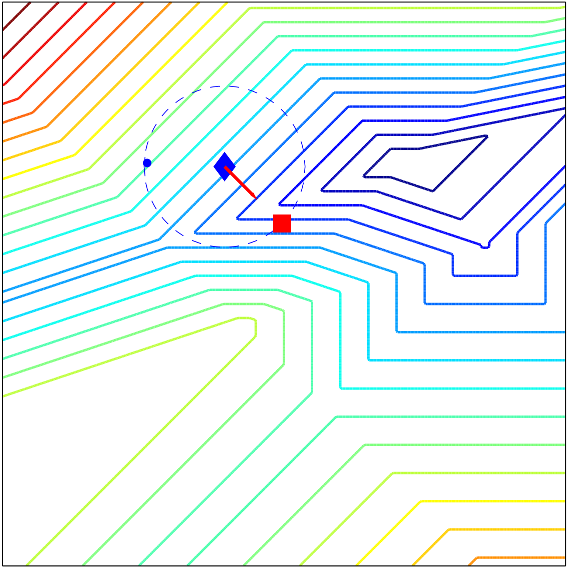
Nonsmooth, composite optimization



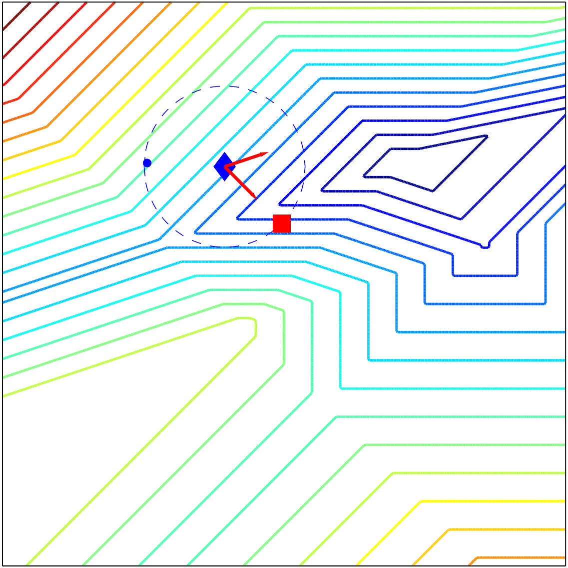
Nonsmooth, composite optimization



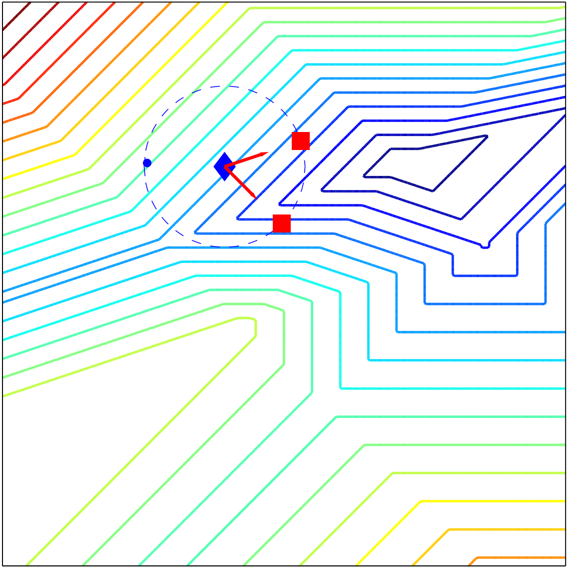
Nonsmooth, composite optimization



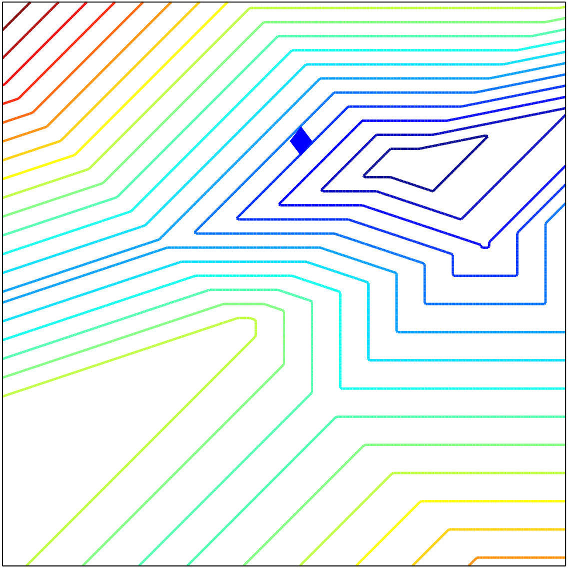
Nonsmooth, composite optimization



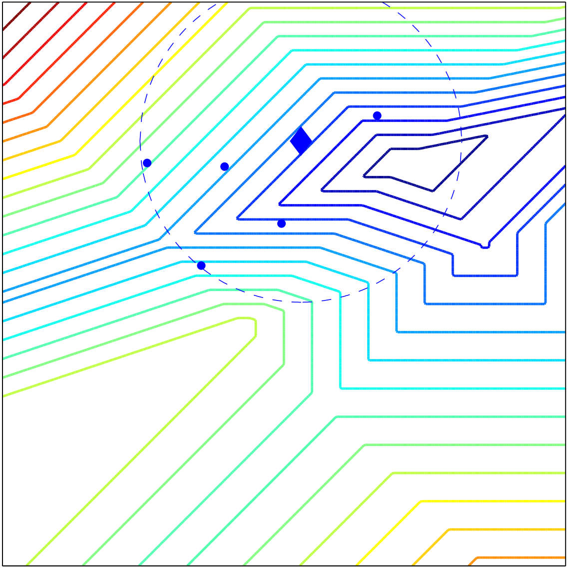
Nonsmooth, composite optimization



Nonsmooth, composite optimization



Nonsmooth, composite optimization



Motivation

- We want to identify distinct, "high-quality", local minimizers of

$$\begin{aligned} \text{minimize } & f(x) \\ l \leq x & \leq u \\ x & \in \mathbb{R}^n \end{aligned}$$

- High-quality can be measured by more than the objective.

Motivation

- We want to identify distinct, “high-quality”, local minimizers of

$$\begin{aligned} \text{minimize } & f(x) \\ l \leq x & \leq u \\ x & \in \mathbb{R}^n \end{aligned}$$

- High-quality can be measured by more than the objective.

- Derivatives of f may or may not be available.

Motivation

- We want to identify distinct, “high-quality”, local minimizers of

$$\begin{aligned} \text{minimize } & f(x) \\ l \leq & x \leq u \\ & x \in \mathbb{R}^n \end{aligned}$$

- High-quality can be measured by more than the objective.
- Derivatives of f may or may not be available.
- The simulation f is likely using parallel resources, but it does not utilize the entire machine.

Global DFO

Theorem (Törn and Žilinskas, *Global Optimization*, 1989)

An algorithm converges to the global minimum of any continuous f on a domain $\mathcal{D}$ if and only if the algorithm generates iterates that are dense in $\mathcal{D}$.

Global DFO

Theorem (Törn and Žilinskas, *Global Optimization*, 1989)

An algorithm converges to the global minimum of any continuous f on a domain $\mathcal{D}$ if and only if the algorithm generates iterates that are dense in $\mathcal{D}$.

- Either assume additional properties about the problem
  - convex f
  - separable f
  - finite domain $\mathcal{D}$

Global DFO

Theorem (Törn and Žilinskas, *Global Optimization*, 1989)

An algorithm converges to the global minimum of any continuous f on a domain $\mathcal{D}$ if and only if the algorithm generates iterates that are dense in $\mathcal{D}$.

- Either assume additional properties about the problem
  - convex f
  - separable f
  - finite domain $\mathcal{D}$
- Or possibly wait a long time (or forever)

Global DFO

Theorem (Törn and Žilinskas, *Global Optimization*, 1989)

An algorithm converges to the global minimum of any continuous f on a domain $\mathcal{D}$ if and only if the algorithm generates iterates that are dense in $\mathcal{D}$.

- Either assume additional properties about the problem
  - convex f
  - separable f
  - finite domain $\mathcal{D}$
- Or possibly wait a long time (or forever)

The theory can be more than merely checking that a method generates iterates which are dense in the domain.

Global DFO

Theorem (Törn and Žilinskas, *Global Optimization*, 1989)

An algorithm converges to the global minimum of any continuous f on a domain $\mathcal{D}$ if and only if the algorithm generates iterates that are dense in $\mathcal{D}$.

- Either assume additional properties about the problem
  - convex f
  - separable f
  - finite domain $\mathcal{D}$
- Or possibly wait a long time (or forever)

The theory can be more than merely checking that a method generates iterates which are dense in the domain.

An algorithm must trade-off between “refinement” and “exploration”.

Global DFO

Theorem (Törn and Žilinskas, *Global Optimization*, 1989)

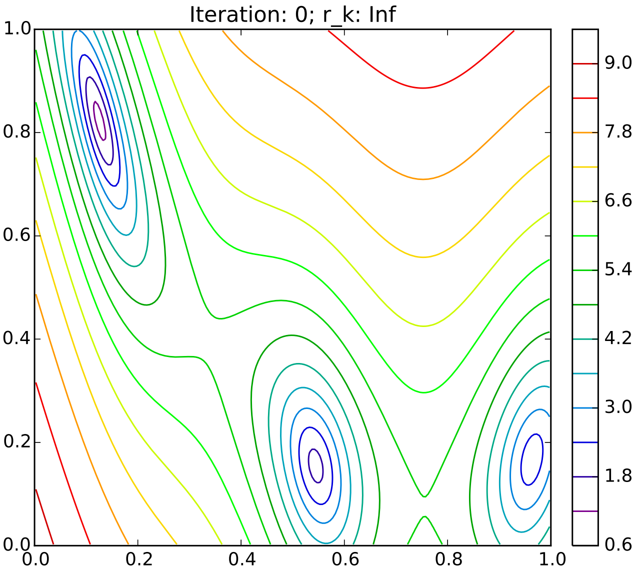
An algorithm converges to the global minimum of any continuous f on a domain $\mathcal{D}$ if and only if the algorithm generates iterates that are dense in $\mathcal{D}$.

- Either assume additional properties about the problem
  - convex f
  - separable f
  - finite domain $\mathcal{D}$
  - concurrent evaluations of f
- Or possibly wait a long time (or forever)

The theory can be more than merely checking that a method generates iterates which are dense in the domain.

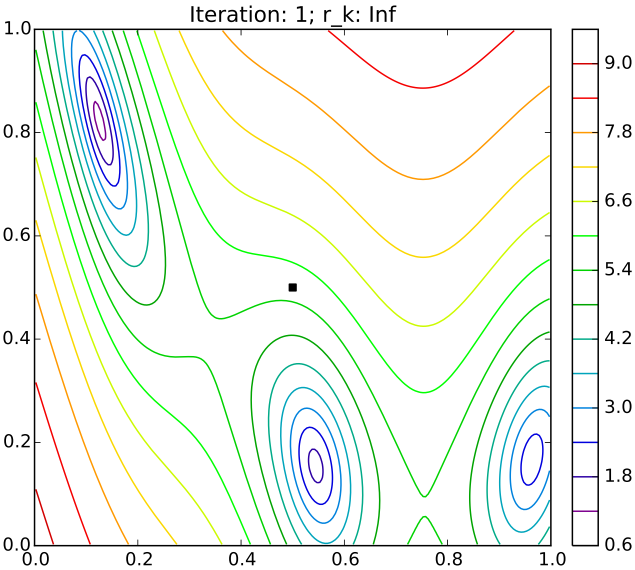
An algorithm must trade-off between “refinement” and “exploration”.

DIRECT



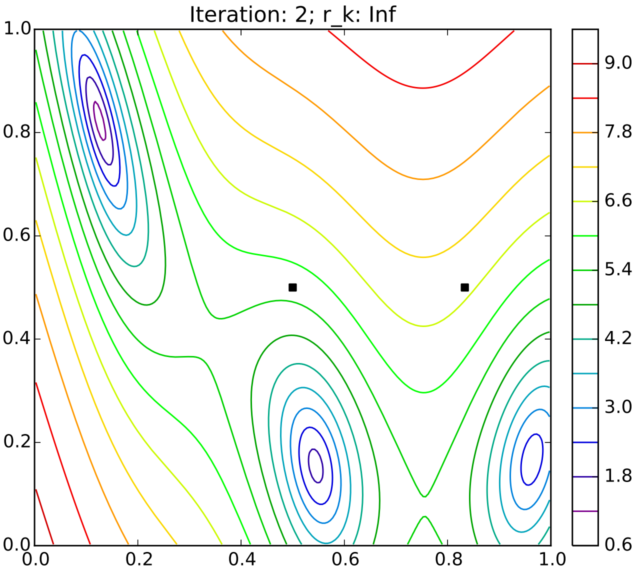
Iteration: 0; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT

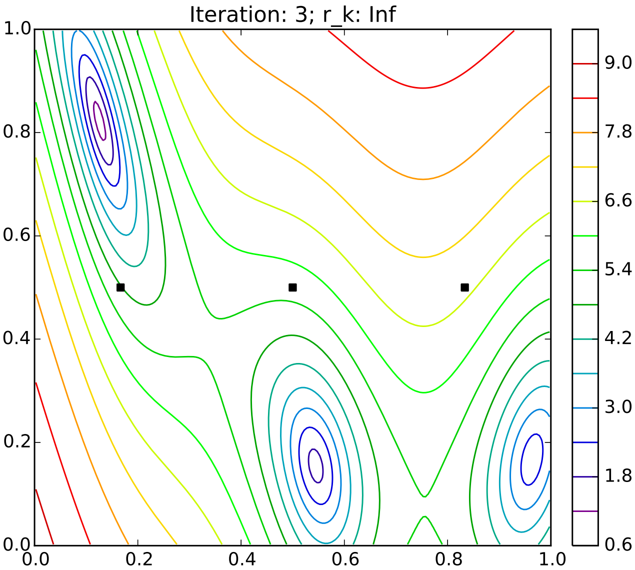


Iteration: 1; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT

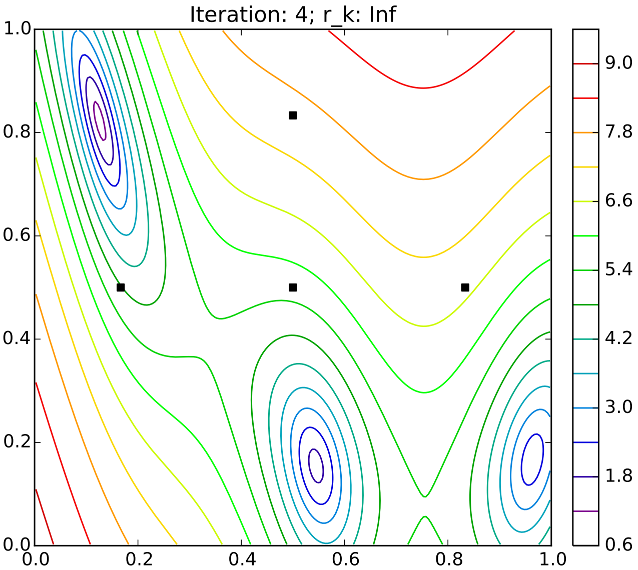

Iteration: 2; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



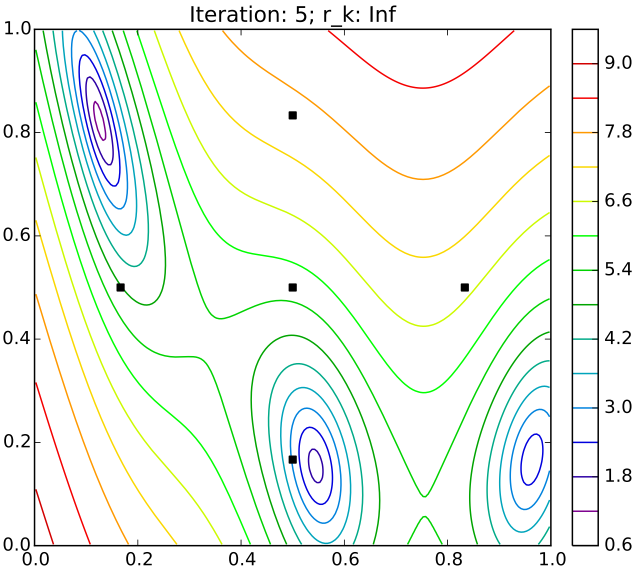
Iteration: 3; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



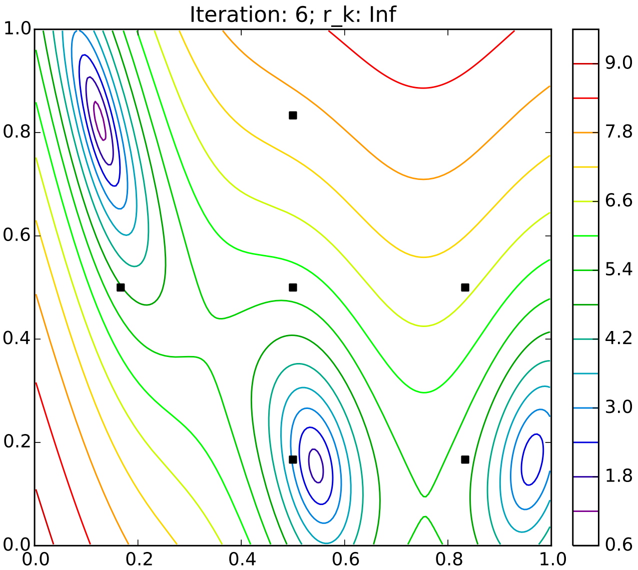
Iteration: 4; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



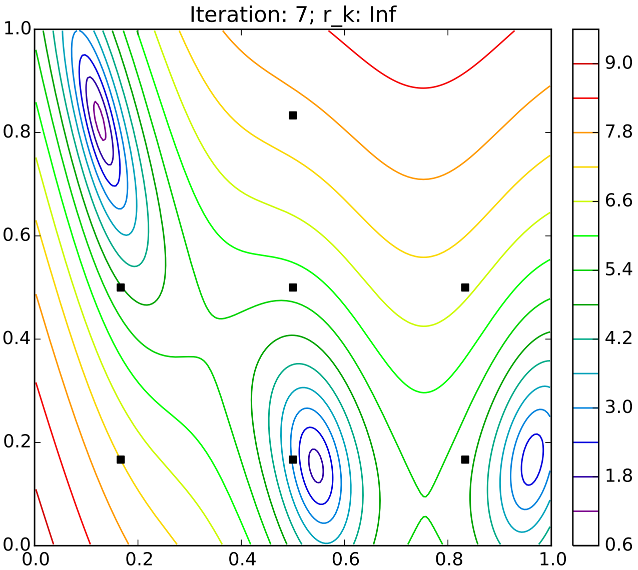
Iteration: 5; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



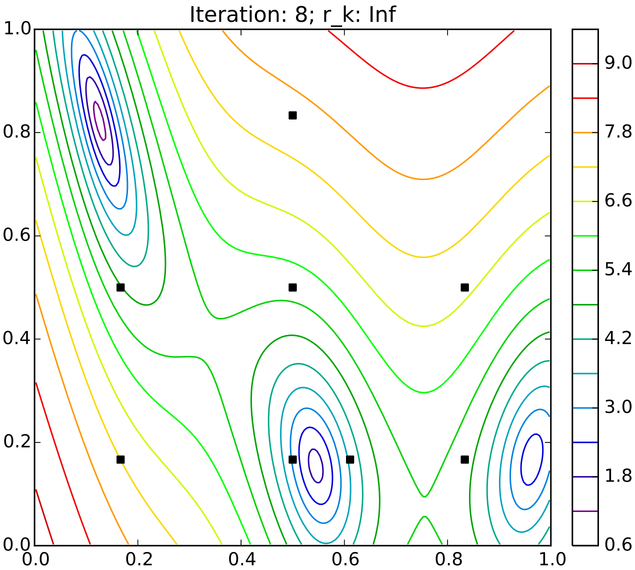
Iteration: 6; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



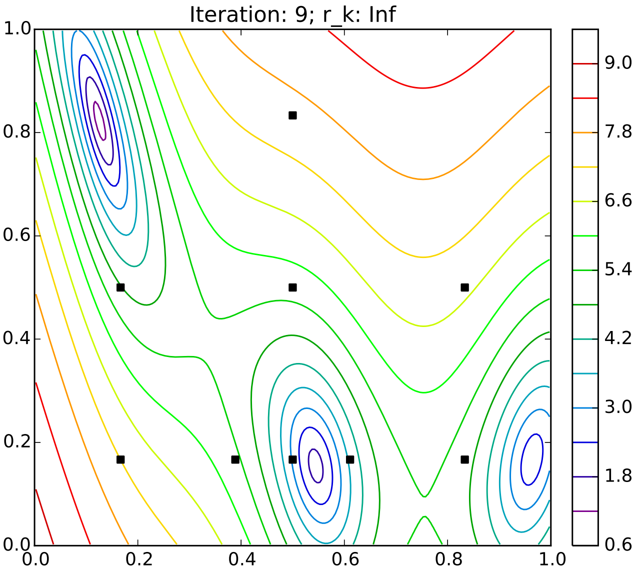
Iteration: 7; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



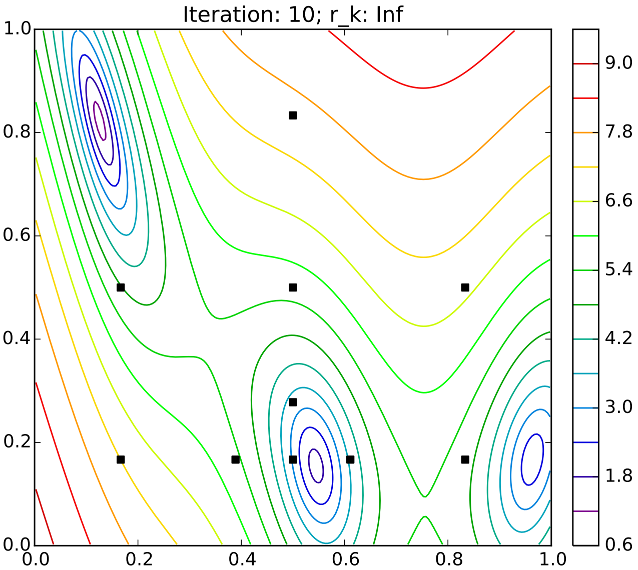
Iteration: 8; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



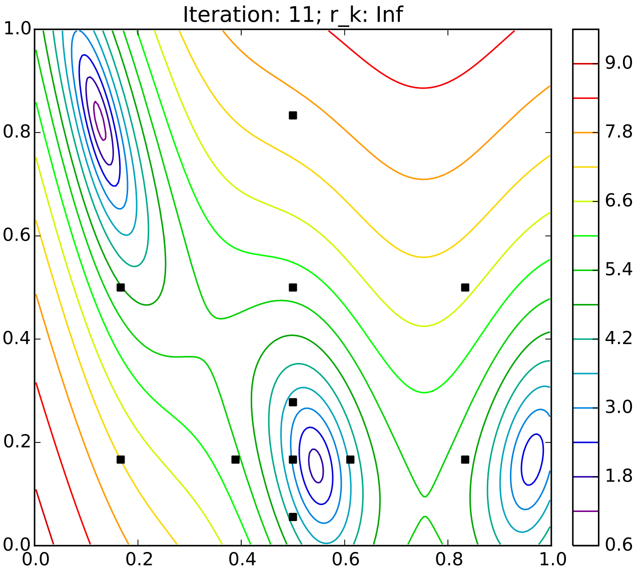
Iteration: 9; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



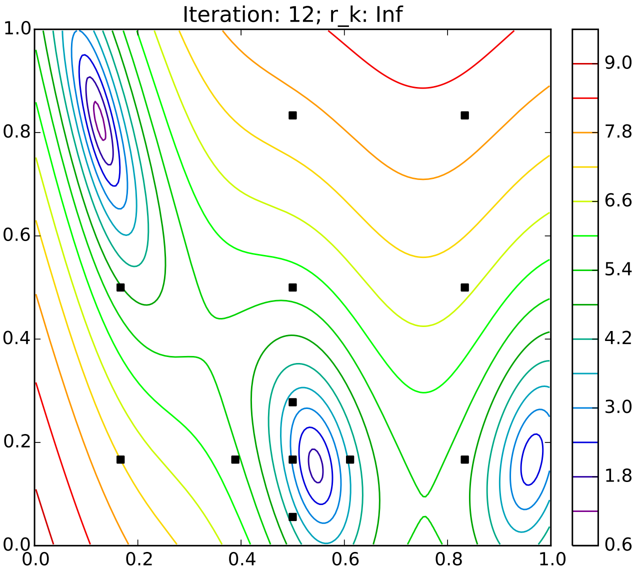
Iteration: 10; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



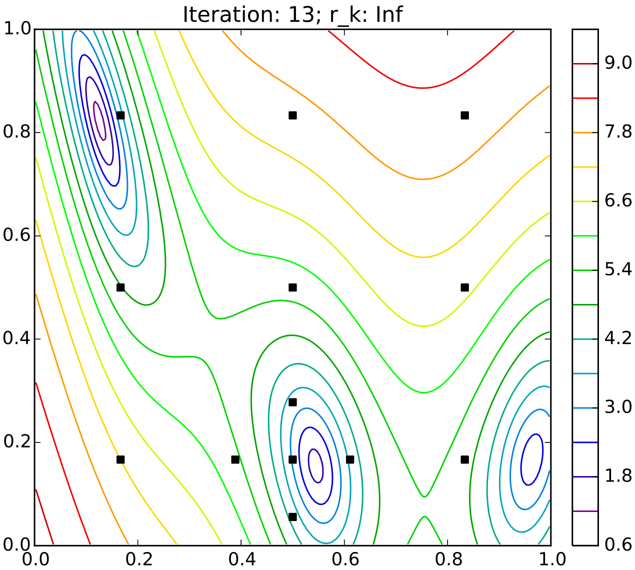
Iteration: 11; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



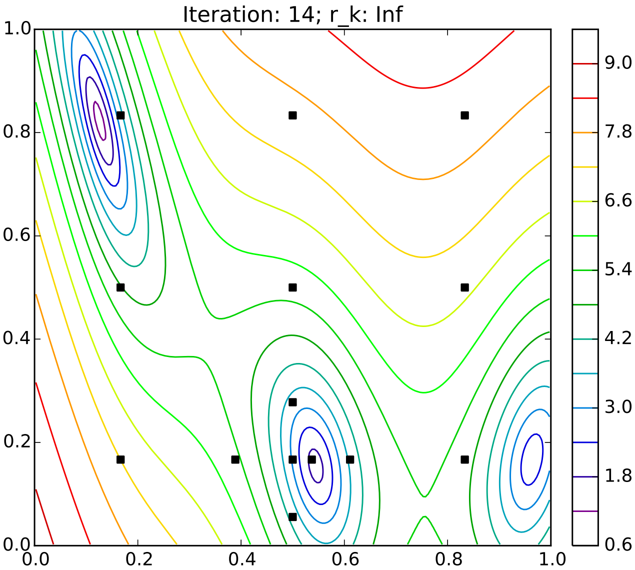
Iteration: 12; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



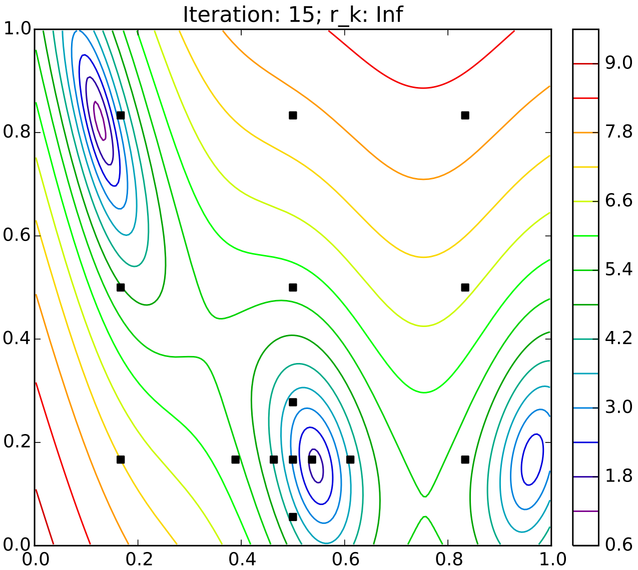
Iteration: 13; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



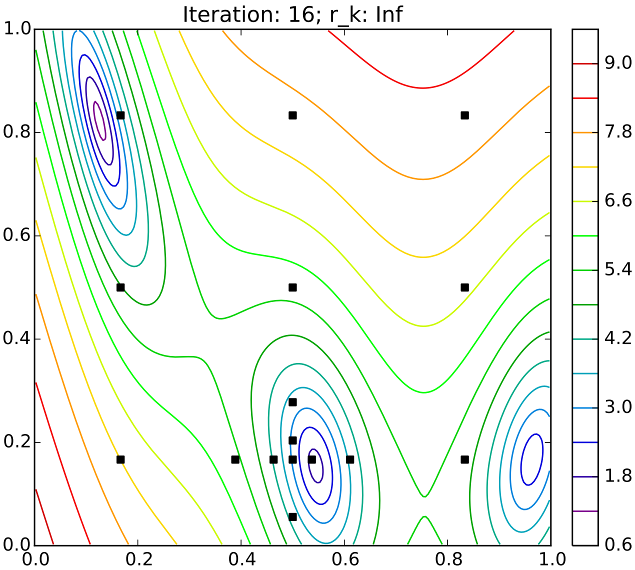
Iteration: 14; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



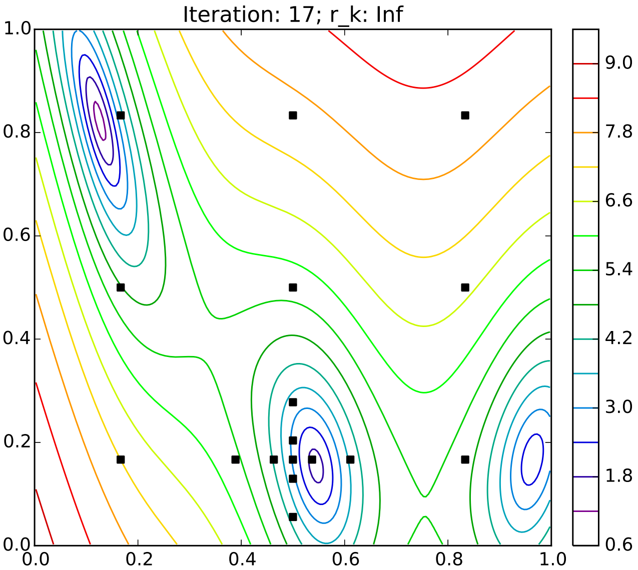
Iteration: 15; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



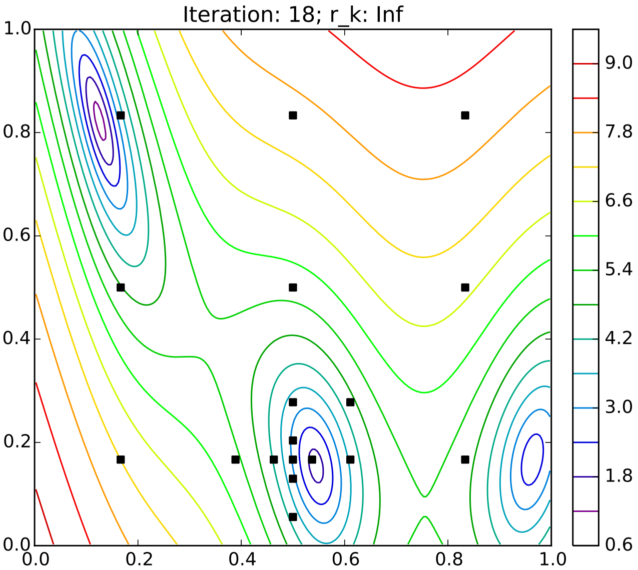
Iteration: 16; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



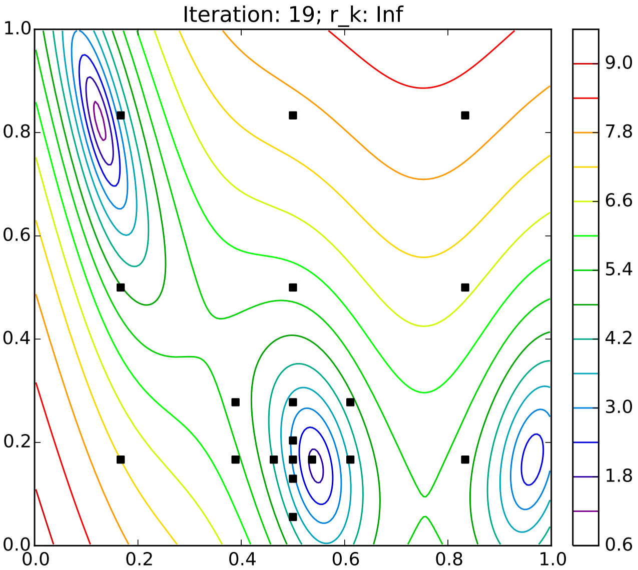
Iteration: 17; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



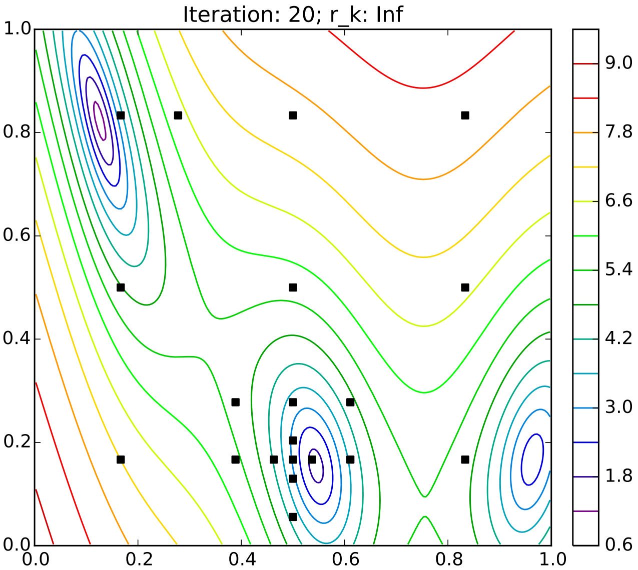
Iteration: 18; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



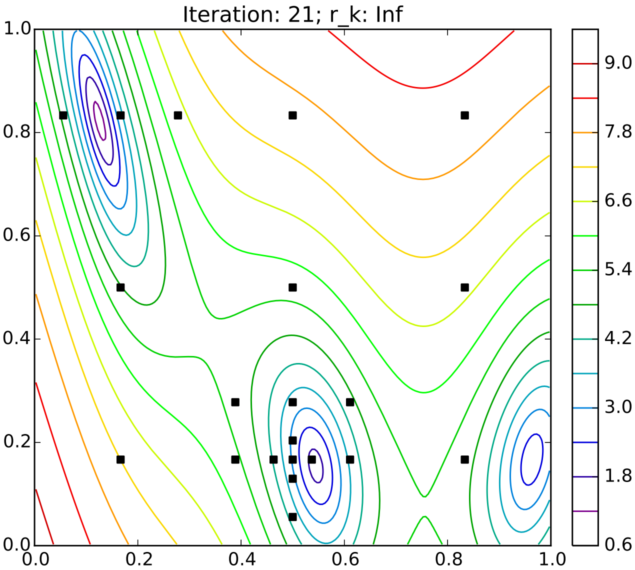
Iteration: 19; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



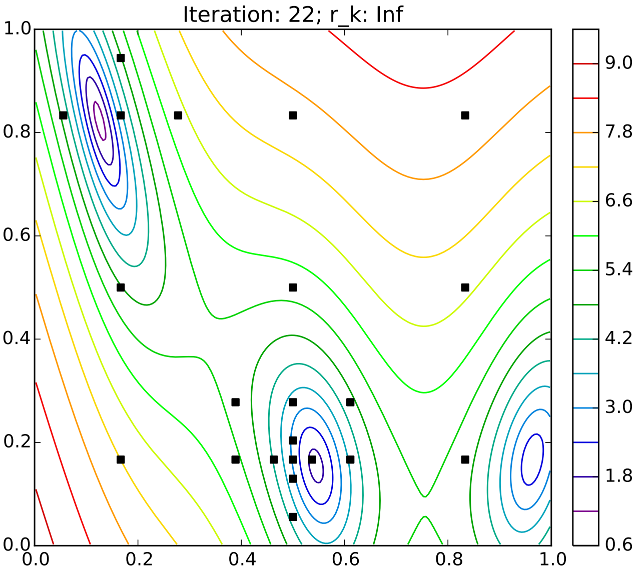
Iteration: 20; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



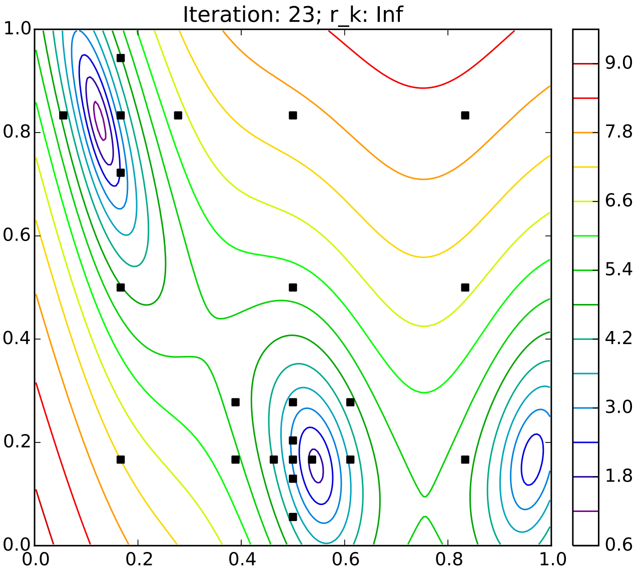
Iteration: 21; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



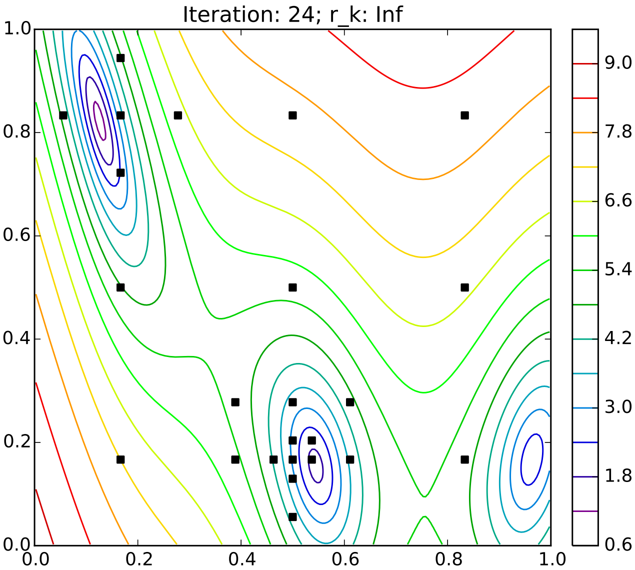
Iteration: 22; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT

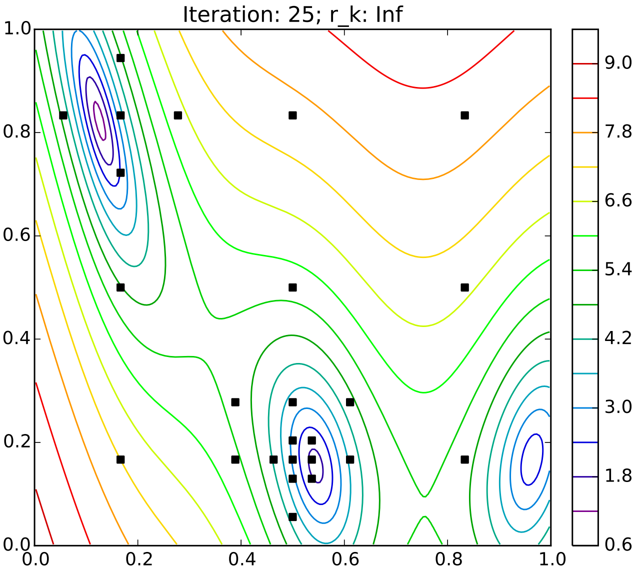


Iteration: 23; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT

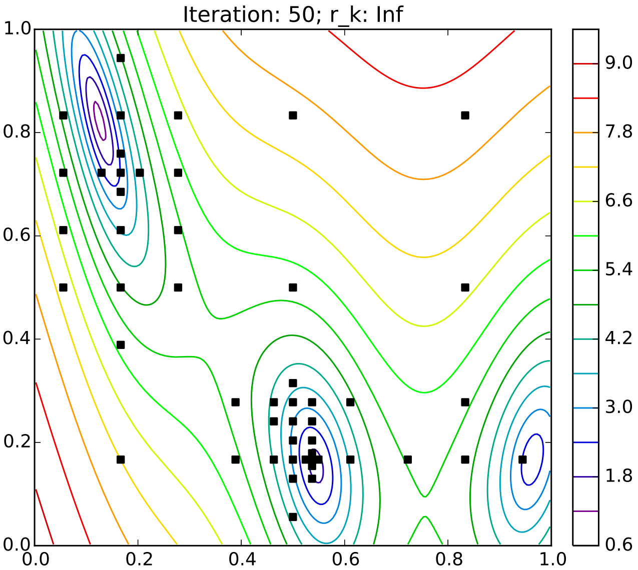

Iteration: 24; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



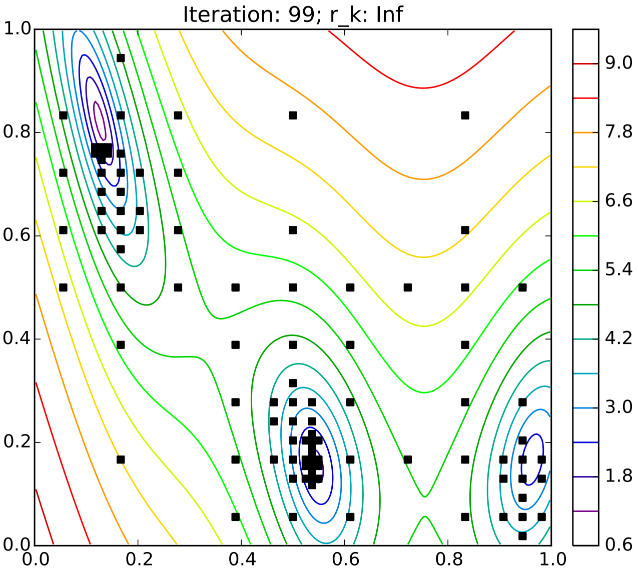
Iteration: 25; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT



Iteration: 50; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

DIRECT


Iteration: 99; r_k: Inf
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
9.0
7.8
6.6
5.4
4.2
3.0
1.8
0.6

Multistart Methods

- Explore by random sampling from the domain $\mathcal{D}$
- Refine by using a local optimization run from some subset of points

Multistart Methods

- Explore by random sampling from the domain $\mathcal{D}$
- Refine by using a local optimization run from some subset of points

Desire to find all minima but start only one run for each minimum

Multistart Methods

- Explore by random sampling from the domain $\mathcal{D}$
- Refine by using a local optimization run from some subset of points

Desire to find all minima but start only one run for each minimum

+ Get to use (more developed) local optimization routines.
  - least-squares objectives, nonsmooth objectives, (un)relaxable constraints, and more
+ Increased opportunity for parallelism
  - objective, local solver, and global solver
- Can require many sequential evaluations for the local solver

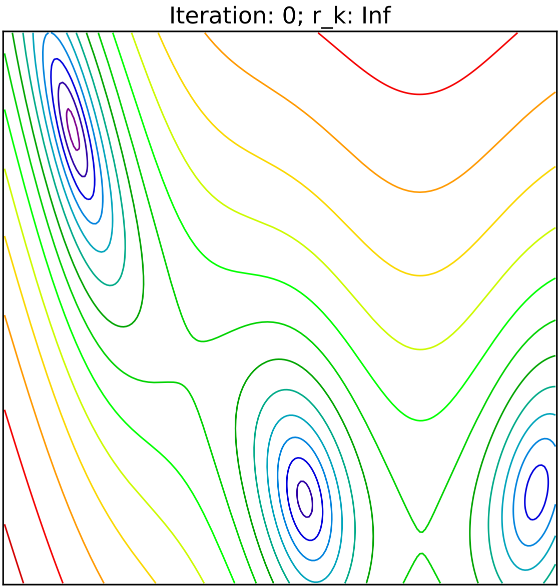
Multistart Theory

- $f \in C^2$, with local minima in the interior of $\mathcal{D}$, and the distance between these minima is bounded away from zero.
- $\mathcal{L}$ is strictly descent and converges to a minimum (not a stationary point).
- $$r_k = \frac{1}{\sqrt{\pi}} \sqrt[n]{\Gamma\left(1 + \frac{n}{2}\right) \mathrm{vol}\left(\mathcal{D}\right) \frac{5 \log kN}{kN}} \tag{1}$$

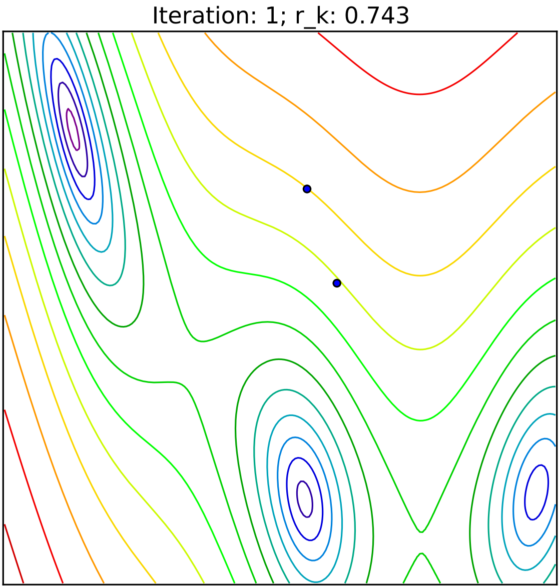
Theorem

If r_k is defined by (1), even if the sampling continues forever, the total number of local searches started is finite almost surely.

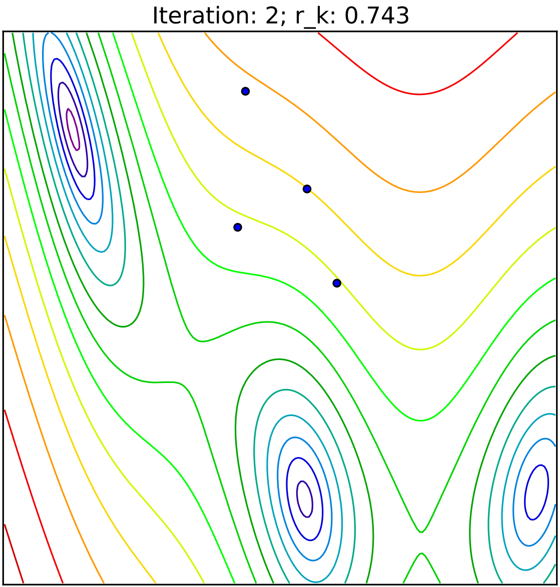
APOSMM



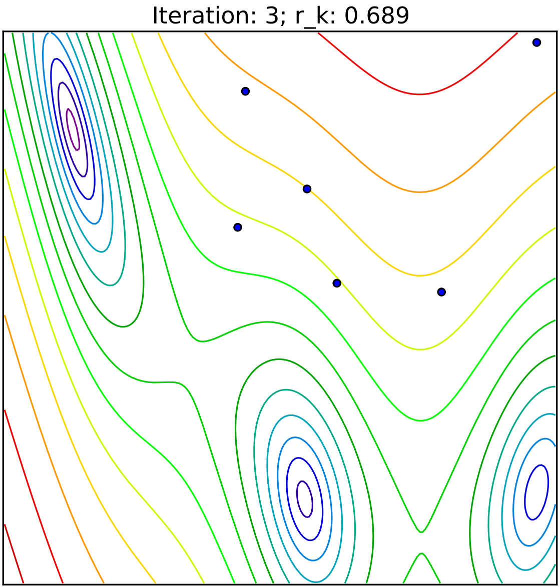
APOSMM



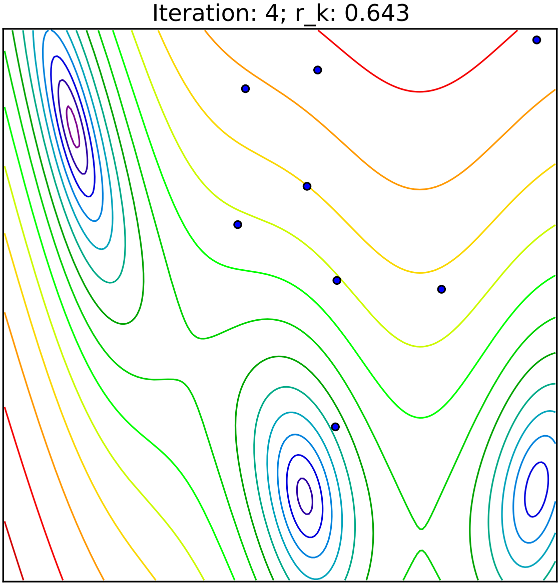
APOSMM



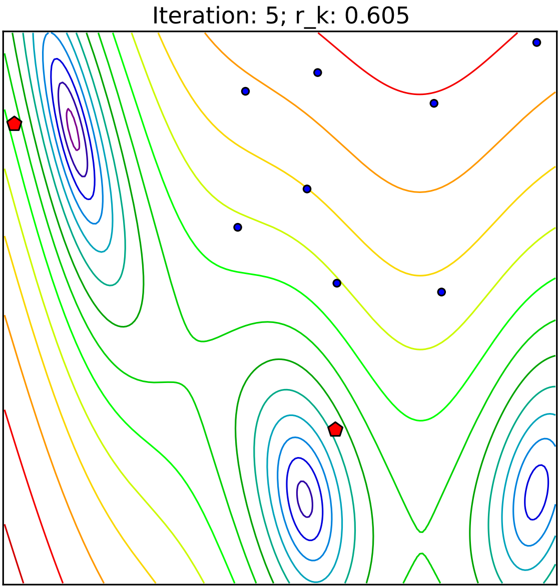
APOSMM



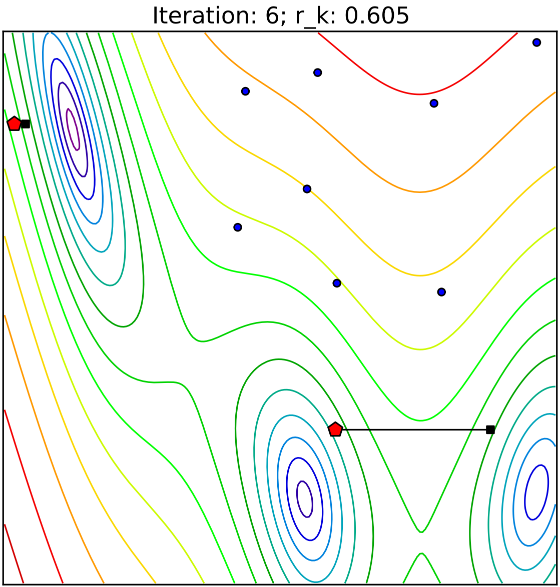
APOSMM



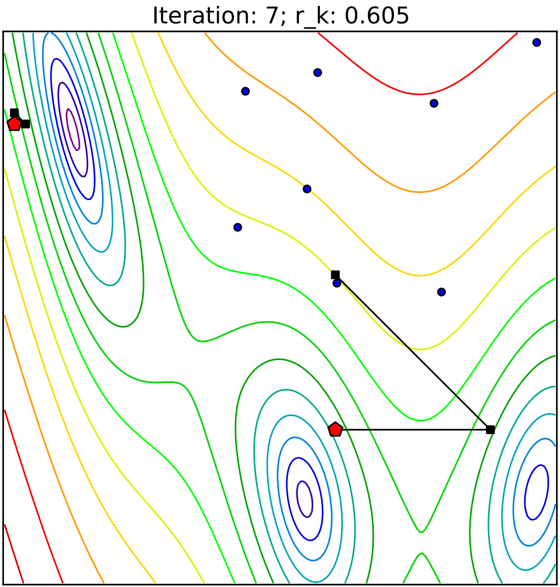
APOSMM



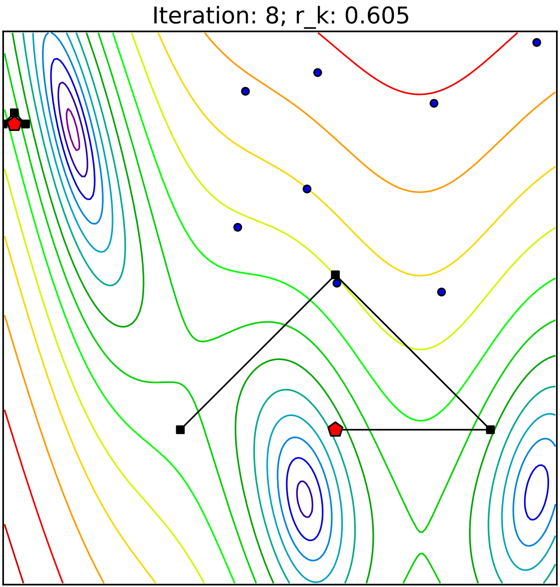
APOSMM



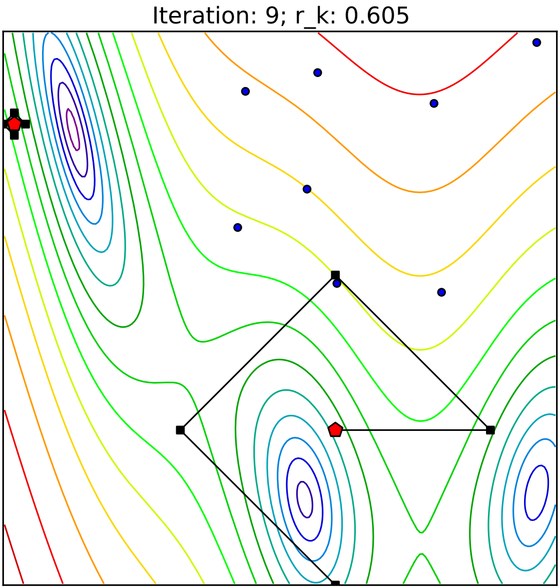
APOSMM



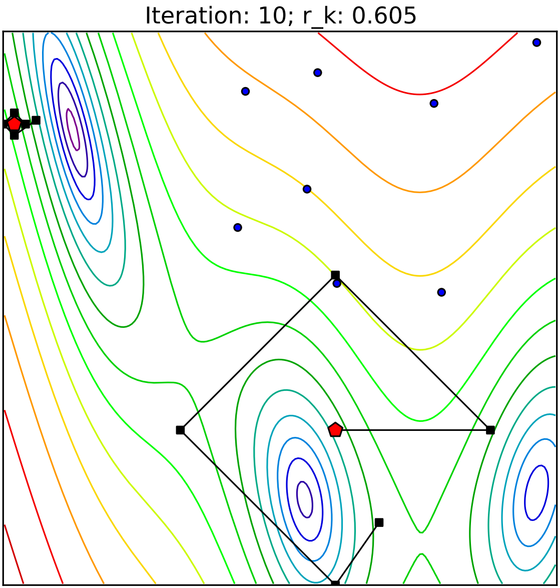
APOSMM



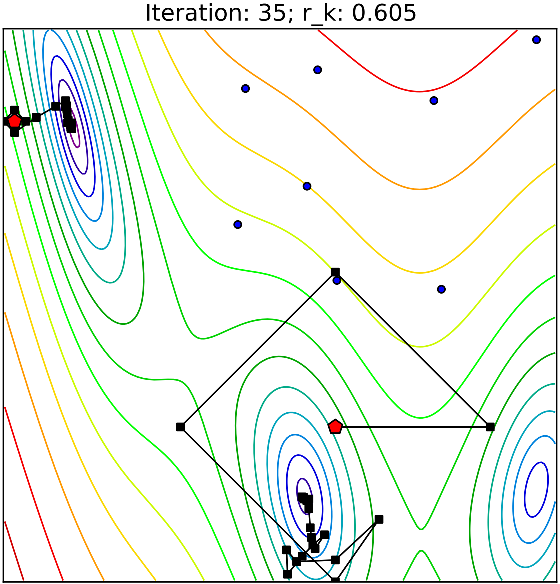
APOSMM



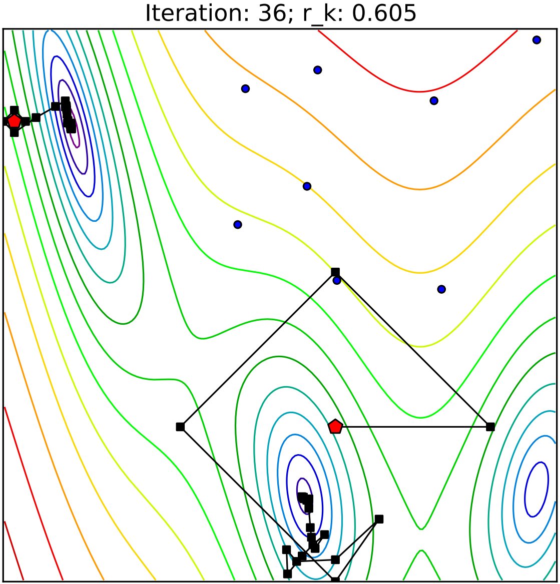
APOSMM



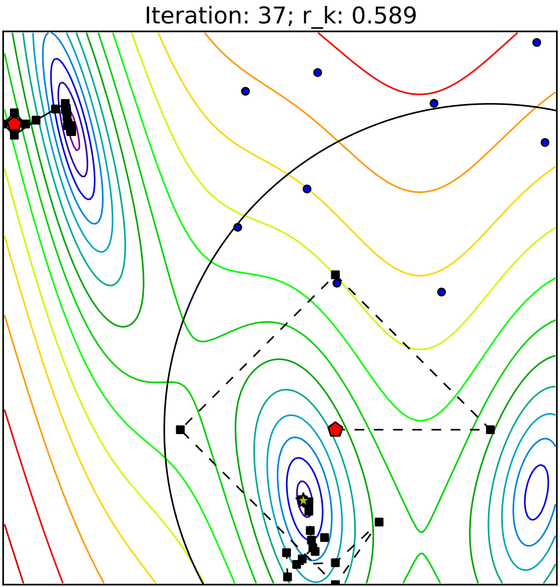
APOSMM



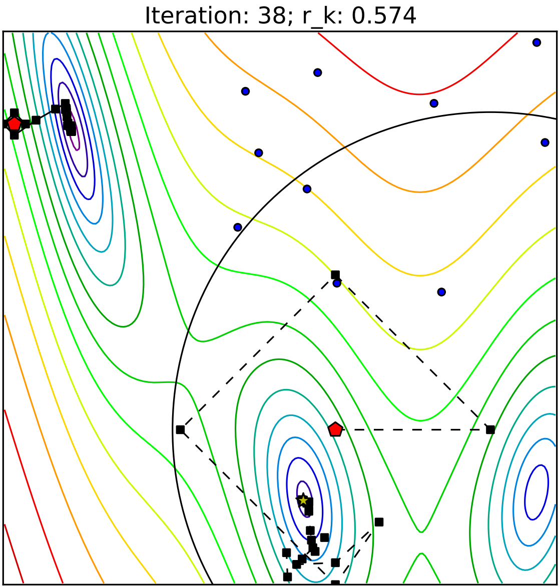
APOSMM



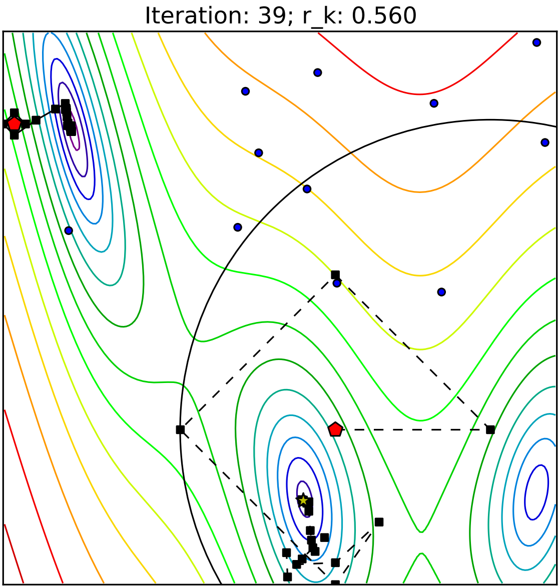
APOSMM



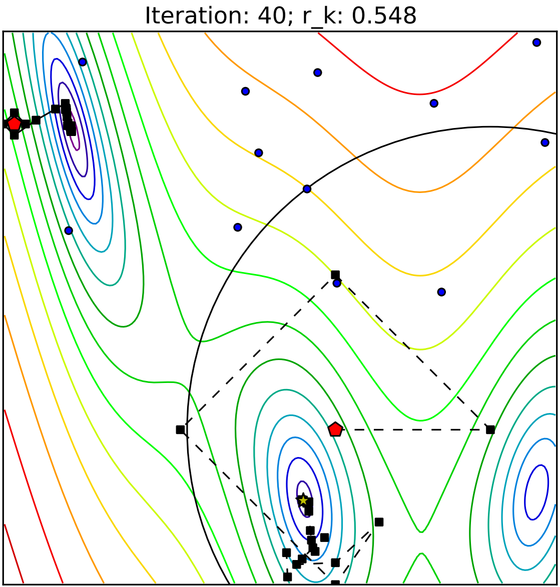
APOSMM



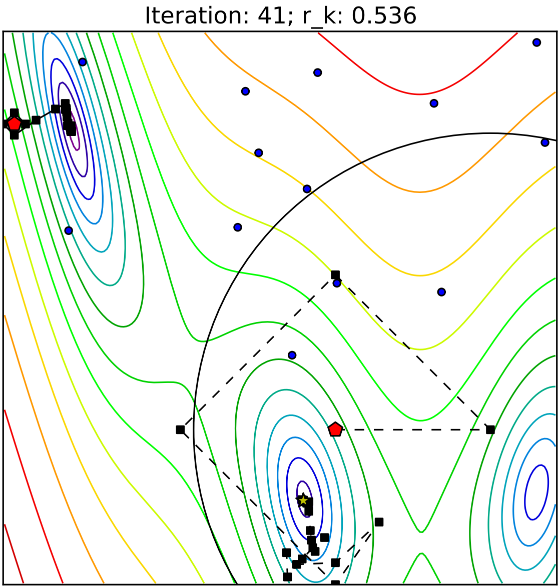
APOSMM



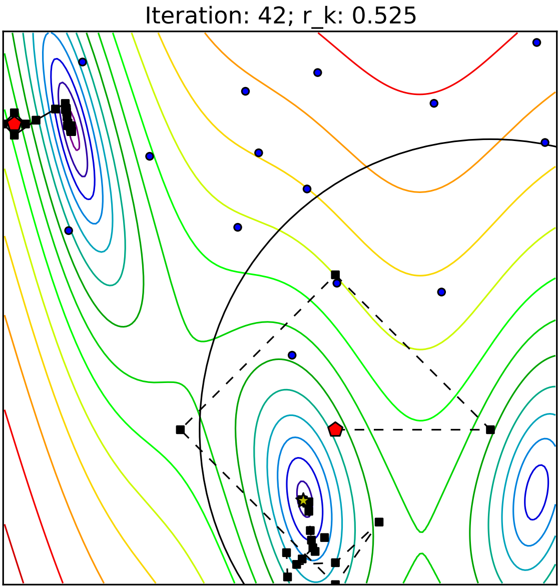
APOSMM



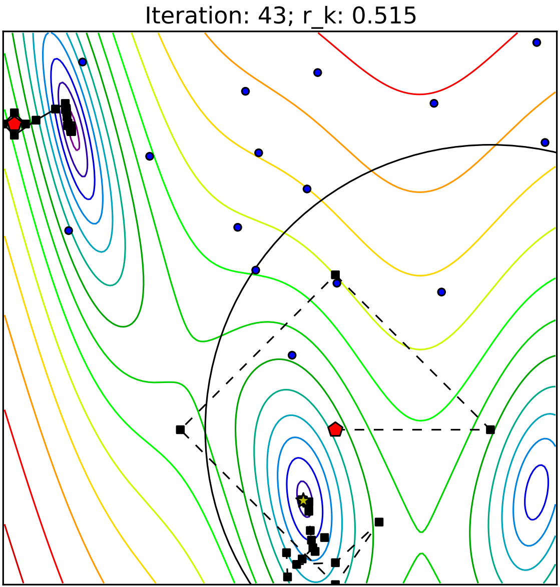
APOSMM



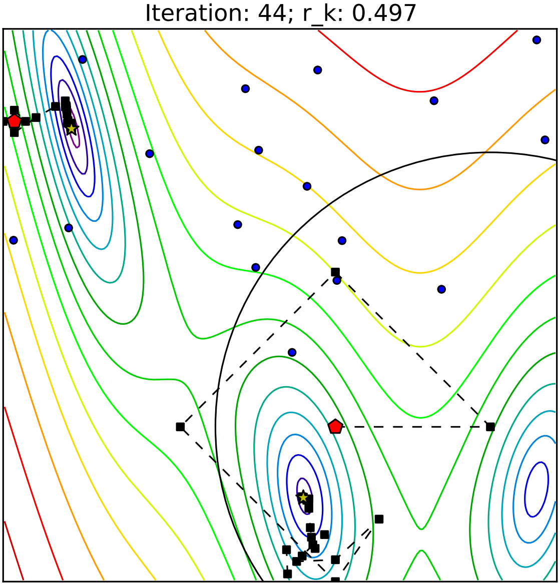
APOSMM



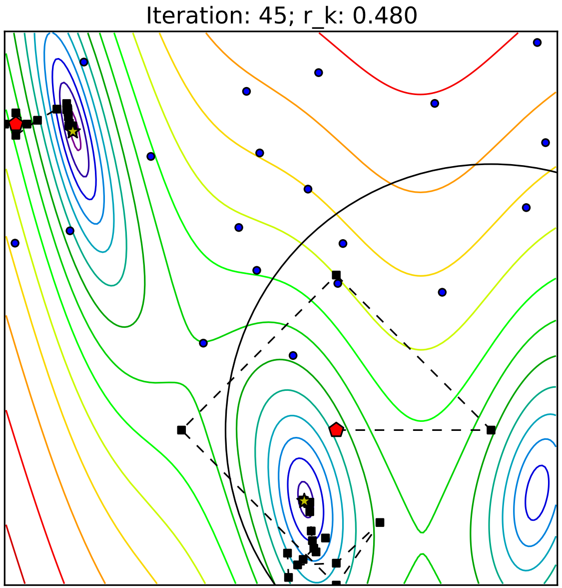
APOSMM



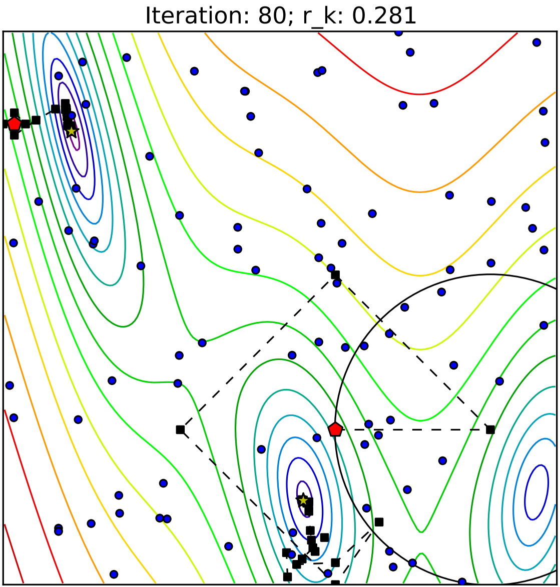
APOSMM



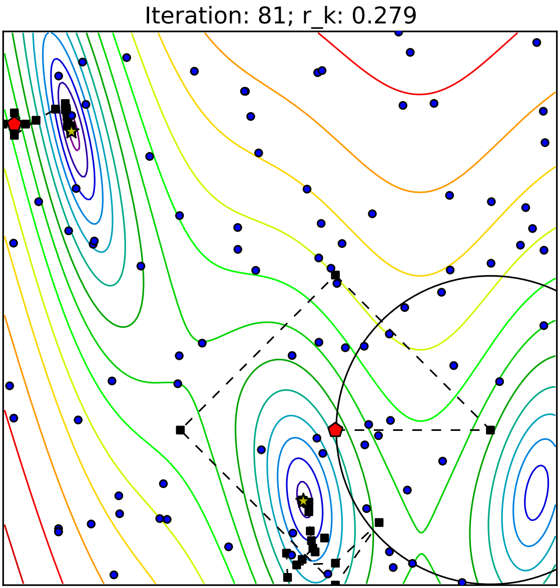
APOSMM



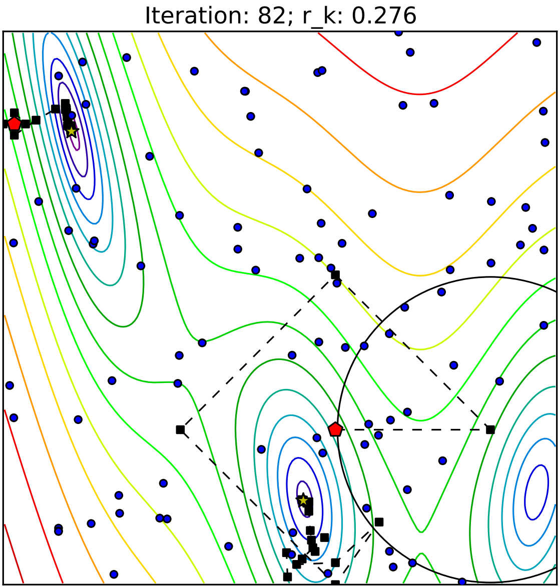
APOSMM



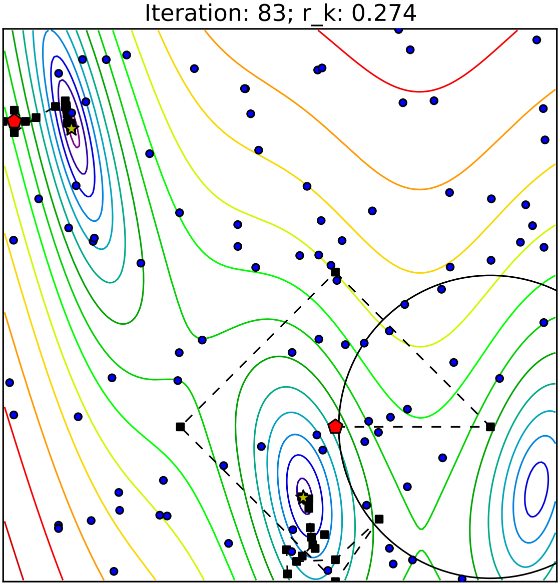
APOSMM



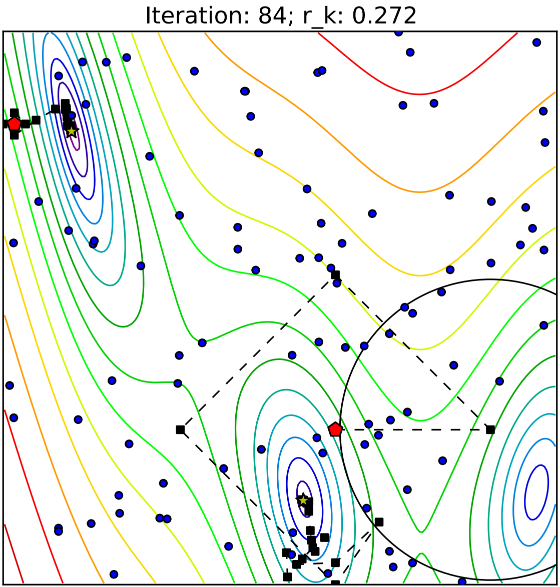
APOSMM



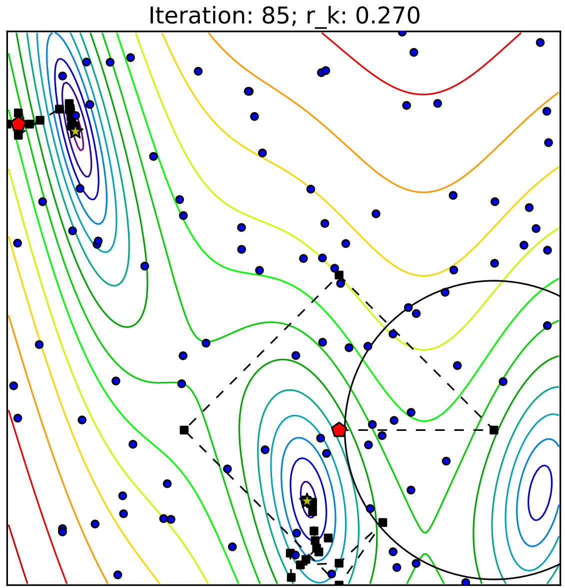
APOSMM



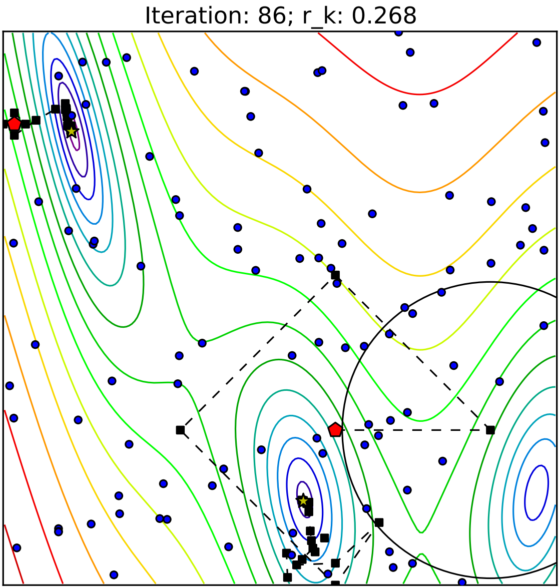
APOSMM



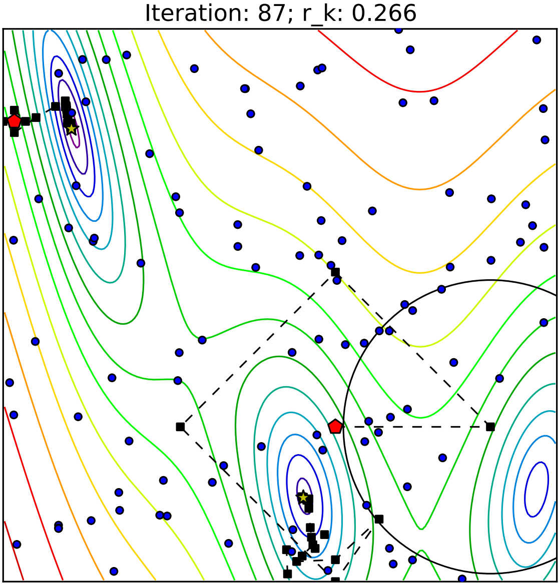
APOSMM



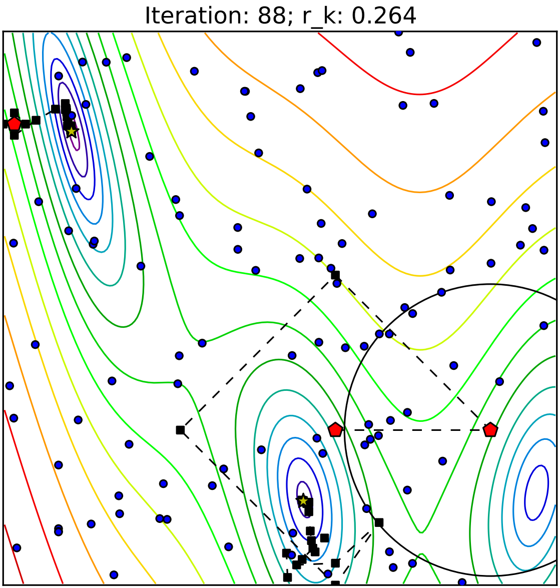
APOSMM



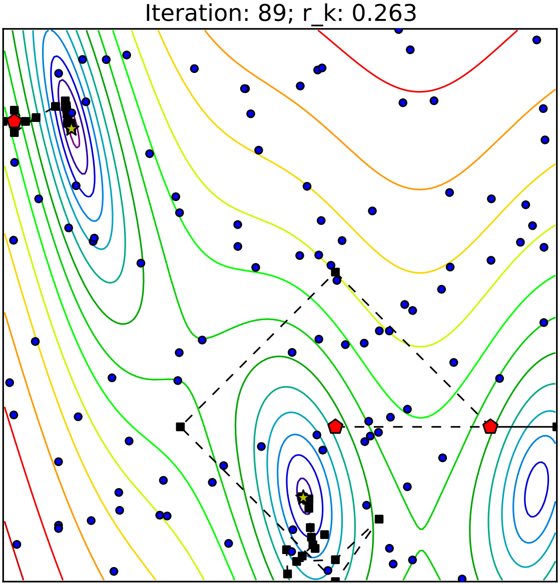
APOSMM



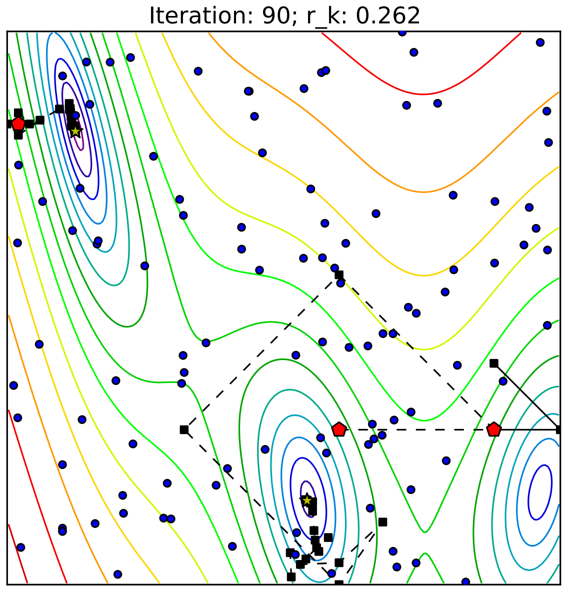
APOSMM



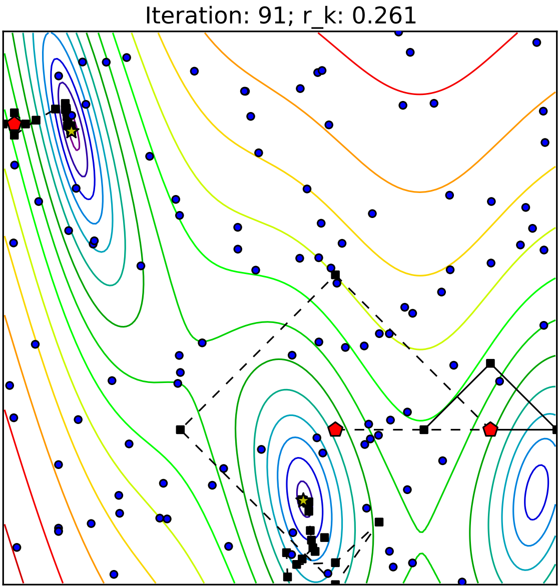
APOSMM



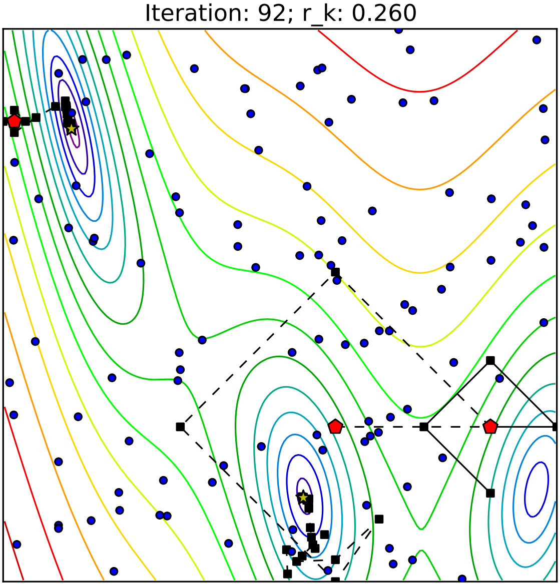
APOSMM



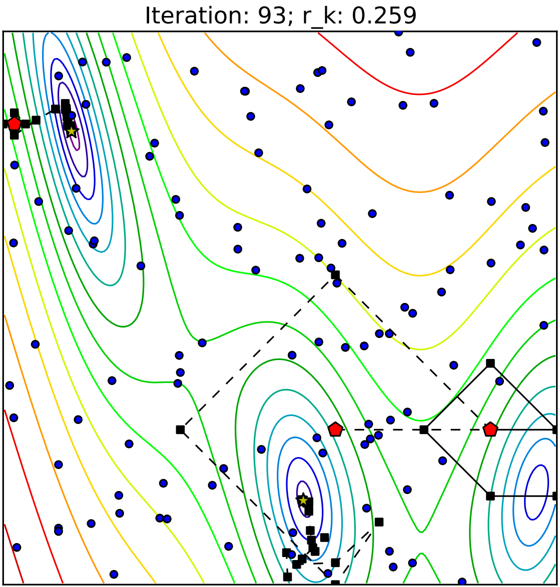
APOSMM



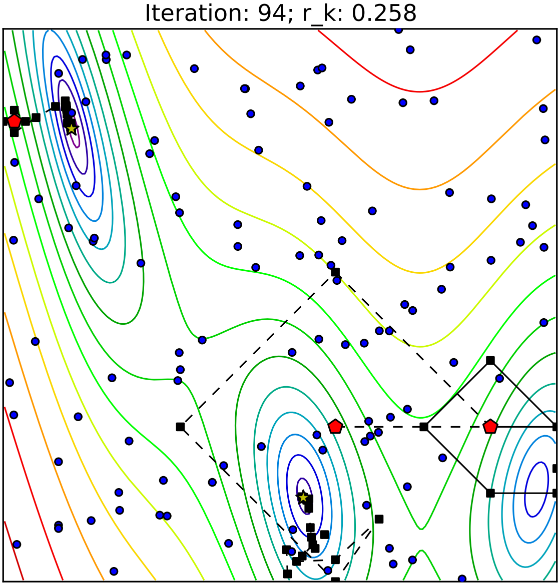
APOSMM



APOSMM

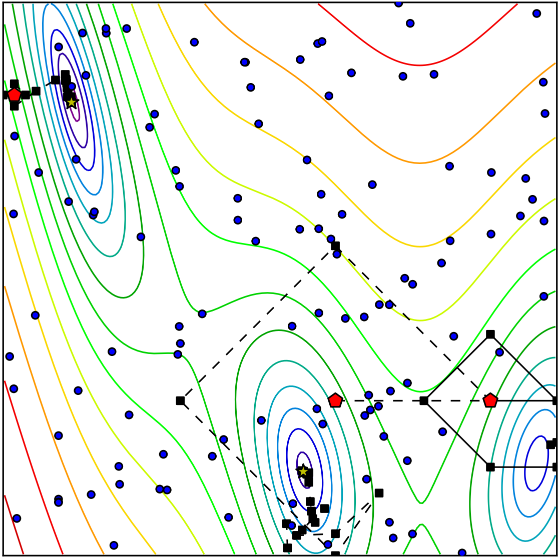


APOSMM

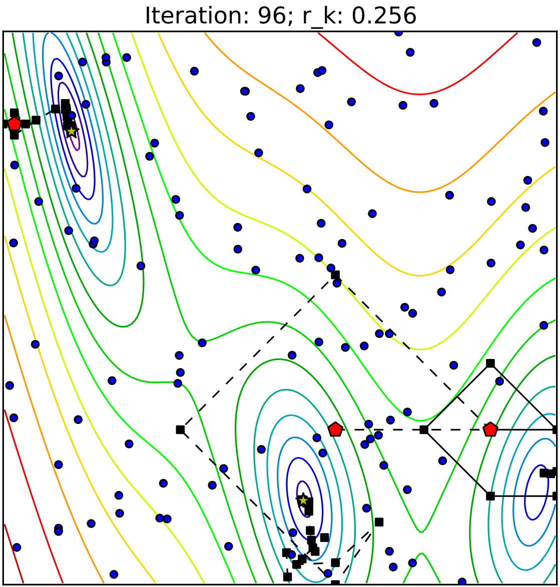


APOSMM

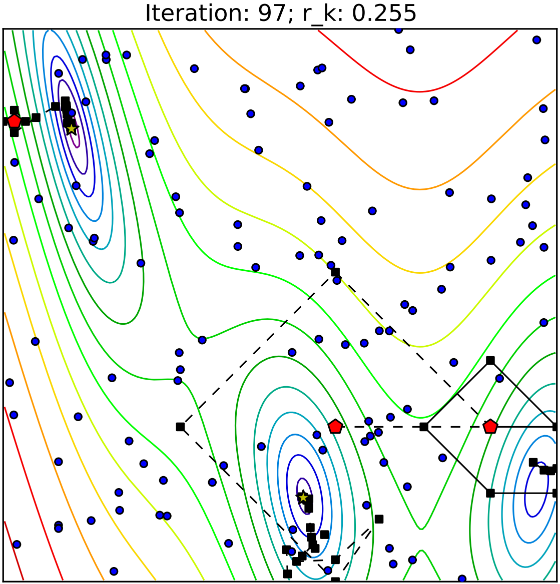
Iteration: 95; r_k: 0.257



APOSMM

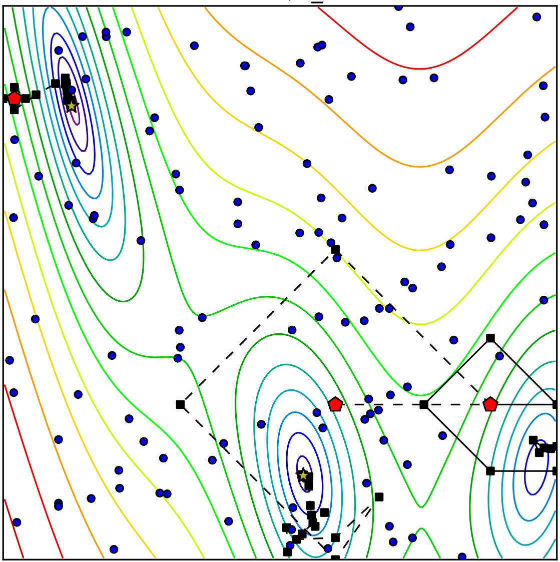


APOSMM

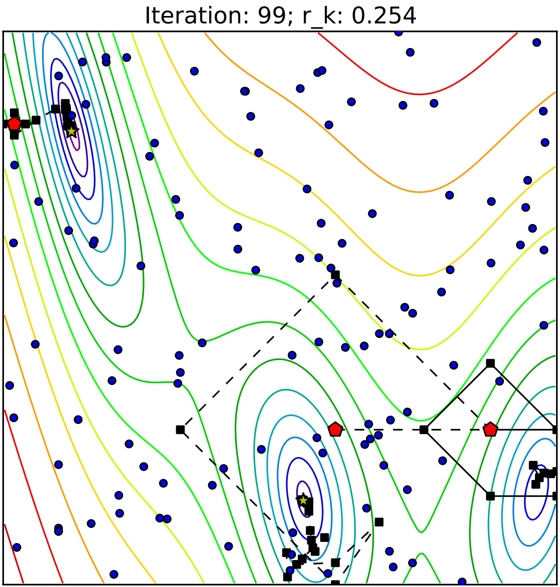


APOSMM

Iteration: 98; r_k: 0.255



APOSMM



DFO warnings

- Be careful
  1) A problem can be written as a scalar output, black box
  2) An algorithm exists to optimize a scalar output, black box function

  1) and 2) true doesn't mean the algorithm should be used

$$\underset{x}{\text{minimize}}\, f(x) = \|Ax - b\|$$

- If your problem has derivatives, please use them. If you don't have them...
  - Algorithmic Differentiation (AD) is wonderful

- Does the problem have structure? Avoid black boxes

Scientific Motivation

- Simulations that need optimization

Scientific Motivation

- Simulations that need optimization
- Simulations are computationally expensive and noisy

Scientific Motivation

- Simulations that need optimization
- Simulations are computationally expensive and noisy
- Want more than just a local minimum

Scientific Motivation

- Simulations that need optimization
- Simulations are computationally expensive and noisy
- Want more than just a local minimum
- Multiple evaluations are often possible

Scientific Motivation

- Simulations that need optimization
- Simulations are computationally expensive and noisy
- Want more than just a local minimum
- Multiple evaluations are often possible
- Evaluation times vary

Scientific Motivation

- Simulations that need optimization
- Simulations are computationally expensive and noisy
- Want more than just a local minimum
- Multiple evaluations are often possible
- Evaluation times vary
- Nonsmooth functions (of the simulation)

Scientific Motivation

- Simulations that need optimization
- Simulations are computationally expensive and noisy
- Want more than just a local minimum
- Multiple evaluations are often possible
- Evaluation times vary
- Nonsmooth functions (of the simulation)
- Stochastic simulations

Scientific Motivation

- Simulations that need optimization
- Simulations are computationally expensive and noisy
- Want more than just a local minimum
- Multiple evaluations are often possible
- Evaluation times vary
- Nonsmooth functions (of the simulation)
- Stochastic simulations
- Multiple objectives (e.g., operational cost and collision energy)

Scientific Motivation

- Simulations that need optimization
- Simulations are computationally expensive and noisy
- Want more than just a local minimum
- Multiple evaluations are often possible
- Evaluation times vary
- Nonsmooth functions (of the simulation)
- Stochastic simulations
- Multiple objectives (e.g., operational cost and collision energy)
- Computational cost that is a function of some variable(s)